ERP/1: Документи

Політики

Технічних вимог, Техноробочого проєкту
і Технічної документації

Зміст

*

Анотація

У статті надається звіт аудиту двох міністерств МВС і МОЗ, де показуються універсальні засади політик проектів для державних IT підприємств. Типовий зміст технічних вимог (з повною таксономією підрозділів, адаптованою на основі Технічних вимог до МВС МІА Документобіг і МОЗ НСЗУ ЕСОЗ продуктів) та структуру технічного завдання відповідно до ДСТУ. Показано принцип як політики визначають вимоги, що узгоджують технічне завдання та посилання на ключові державні стандарти документування ДСТУ. Матеріал призначений для архітекторів, аналітиків та розробників, які працюють над державними та комерційними проєктами в Україні.

Політики

Політики є первинним і незмінним документом будь-якого проєкту. Вони визначають правила гри до того, як формулюються вимоги чи обираються технології.

Загальні принципи

1. Усі рішення в проєкті приймаються виключно на основі затверджених політик.
2. Політики є первинними документами. Технічні вимоги формуються виключно як похідні від політик.
3. Технічне завдання є єдиним документом, де дозволяється вказувати конкретні продукти, технології та вендорів.
4. Архітектура повинна залишатися агностичною до моменту затвердження Технічного завдання.

Технічні вимоги

Перелік технічних вимог наведених в політиках

- Функціональні вимоги до словникової системи
- Функціональні вимоги до набору облікових сутностей і їх операцій
- Функціональні вимоги до набору валідацій
- Функціональні вимоги до рольової моделі
- Функціональні вимоги до інтерфейсу користувача
- Функціональні вимоги до системи процесів
- Нефункціональні вимоги до архітектури
- Нефункціональні вимоги до безпеки
- Нефункціональні вимоги до потужності та ємності
- Нефункціональні вимоги до системи логування
- Нефункціональні вимоги до інтерфейсу користувача
- Нефункціональні вимоги до контролю якості

Повний зміст технічних вимог

Структура технічних вимог узагальнена для будь-якого високотехнологічного державного ПЗ.

- Умовні скорочення та визначення
- Загальні відомості
  - Передумови
  - Питання, що вирішуються
  - Вимоги законодавства та міжнародних стандартів (ДСТУ, ISO, NIST, ITU-T)
- Призначення та цілі впровадження
- Класифікація вимог
  - 3.1. Функціональні вимоги
    - 3.1.1. Опис довідників
    - 3.1.2. Опис реєстрових сутностей та їх станів
    - 3.1.3. Опис бізнес-процесів (BPMN)
    - 3.1.4. Вимоги до протоколів взаємодії
    - 3.1.5. Опис прав та привілеїв (ABAC/RBAC/ACL)
    - 3.1.6. Опис валідацій
    - 3.1.7. Шаблони сповіщень
    - 3.1.8. Вимоги до серіалізації
    - 3.1.9. Вимоги до транспорту
  - 3.2. Нефункціональні вимоги
    - 3.1.1. Вимоги до архітектури
    - 3.2.1. Вимоги до безпеки
    - 3.2.2. Вимоги до потужності
    - 3.2.4. Вимоги до функціональності логування
    - 3.2.5. Адміністративні вимоги
      - 3.2.5.1. Настанова до видів тестування
      - 3.2.5.2. Вимоги до настанов адміністратора
      - 3.2.5.3. Вимоги до документування
      - 3.2.5.4. Технологічний стек (агностичний опис)
    - 3.2.6. Загальні вимоги до документації та артефактів
    - 3.2.7. Юридичні вимоги

Технічне завдання

Структура технічного завдання відповідає ДСТУ 3973-2000 «Система розроблення та постачання продукції на виробництво. Правила виконання науково-дослідних робіт. Загальні положення» та ДСТУ 3974-2000 (для конструкторських робіт).

Зміст технічного завдання

ДСТУ 3973-2000 визначає зміст технічного завдання.

1. Мотивація, мета, виконавці
2. Класифікація вимог (функціональні, нефункціональні, технічні, адміністративні, ергономічні, політичні, юридично-правові, документальні, архітектурні)
3. Принципові вимоги (відповідність ДСТУ, NIST, ISO, ITU-T, ДССЗІ)
4. Функціональні вимоги
5. Ергономічні вимоги
6. Технічні вимоги
7. Адміністративні вимоги
8. Політичні вимоги
9. Юридично-правові вимоги
10. Документування (перелік документів)
11. Етапи виконання
12. Результати
13. Порядок передачі

Посилання на стандарти ДСТУ

Стандарт	Назва та призначення
ДСТУ 3973-2000	Система розроблення та постачання продукції. Правила виконання науково-дослідних робіт. Загальні положення (основа структури ТЗ)
ДСТУ 3974-2000	Правила виконання науково-конструкторських робіт
ДСТУ 42010:2018	Інженерія систем і програмних засобів. Опис архітектури
ДСТУ 62264:2019	Інтеграція систем управління підприємством і виробництвом
ДСТУ 3008:2015	Документація на розроблення ПЗ
ДСТУ 3396.0-96, 3396.1-96, 3396.2-97	Комплекс стандартів на уніфіковану систему документації
ДСТУ 4541, ДСТУ 28147	Криптографічні стандарти України
ДСТУ 19.403–19.781, ДСТУ 34.003, 34.201, 34.601–34.602	Obsolete but mandatory for compatibility documentation standards

Посилання на нормативно-правові акти (НПА)

Засоби інформатизації, розробка та впровадження Системи регулюються такими нормативно-правовими актами України:

НПА	Назва та призначення
Закон України № 74/98-ВР	Про Національну програму інформатизації (визначає правові основи планування та виконання проектів)
Постанова КМУ від 21.02.2025 № 205	Деякі питання створення, адміністрування та забезпечення функціонування засобу інформатизації (затверджує Обов'язкові вимоги до засобів інформатизації)
Порядок локалізації програмних продуктів	Затверджений Постановою КМУ № 205 (визначає вимоги до локалізації ПЗ для НПІ)
Постанова КМУ від 12.04.2022 № 436	Про затвердження Положення про Національну програму інформатизації
Закон України № 2155-VIII	Про електронні довірчі послуги
Закон України № 2297-VI	Про захист персональних даних
Закон України № 80/94-ВР	Про захист інформації в інформаційно-телекомунікаційних системах

Функціональні вимоги

Вимоги до процесів системи

Система повинна буди ізоморвна моделі FSM (Finite State Machines), BPMN (ISO 19510), або Мережам Петрі, тобто повинна підтримувати парелальні і послідовні обчислення (погодження) необхідні для орагіназіції документообігу. Визначення процесу, як сутності, що складається з наступних частин (як приклад такої ізоморфної системи):

- Контекст процесу у вигляді переліку документів
- Перелік станів процесу, кожен з яких є робочим місцем користувача або сервісом
- Переходи між станами що є функціями з логікою
- Початковий і кінцевий стани процесу
- Історія кроків процесу з документами

Кожен користувач визначається як сутність, що містить наступні реквізити:

- Сертифікат доступу КЕП X.509
- ЄДРПО підприємства
- Посада, ПІБ, табельний номер
- Визначені ролі

Система бізнес-процесів повинна формально визначати:

- Перелік користувачів і структури підприємства
- Перелік ролей
- Перелік реєстрових сутностей, їх реквізитів і операцій над ними
- Перелік документів і їх реквізитів
- Перелік інформаційних повідомлень і їх реквізитів
- Перелік бізнес-процесів

Вимоги до викликів API

Для початку розробки аналітик замовника передає вендору зазначені верхньорівневі вимоги до методу. Після завершення розробки вендор надає опис методу за затвердженим шаблоном відповідно до настанови. Документація по опису методів API повинна бути створена згідно з затвердженими шаблонами.

- призначення методу: яку дію він має виконувати;
- формулювання методу: метод та ендпойнт;
- які параметри необхідні для виклику методу — початкові умови для застосування методу (параметри мають бути представлені у формі таблиці із зазначеним типом та прикладом формулювання);
- відповідь сервера у формі структурованих даних — інформація у форматі серіалізації (зразок відповіді описують у формі коду);
- обробка помилок: опис ситуації, коли виконання дії неможливе з певних причин (перелік помилок мають бути представлені у формі таблиці з розшифровкою значення);
- додаткова інформація: обмеження доступу, використання стандартів тощо (може бути представлена у формі довільного тексту в окремому розділі).

Вимоги до інтерфейсу користувача

Обробка помилок

Інтерфейс користувача має представляти чіткі та інформативні повідомлення про помилки із зазначенням їхньої причини та шляху виправлення. Інтерфейс користувача має надавати можливість повторної спроби операцій після усунення факторів, які спричинили помилку.

Загальні вимоги

Інтерфейс користувача має відповідати наступним загальним вимогам:

- Відповідність стандартам вебдоступності (WCAG): Забезпечення доступності для користувачів з обмеженими можливостями (наприклад, підтримка клавіатурної навігації, альтернативний текст для зображень, достатній контраст кольорів). Відповідність принаймні рівню AA стандартів WCAG 2.1.
- Наслідування дизайн-коду компанії: Інтерфейс має відповідати існуючому дизайн-коду компанії (стиль, кольори, шрифти, логотипи тощо). Використання компонентів UI з бібліотеки компонентів компанії (якщо така є для певного функціоналу). Забезпечення консистентності візуального стилю між різними частинами системи.

Нефункціональні вимоги

Вимоги до архітектури

Перший підрозділ описує загальну архітектуру ERP/1, та вимоги до розробників додаткової функціональності. Ця глава визначає карту вимог та описує загальний перелік вимог, який випливає з логіки розгортання архітектурних рівнів ISO-42010.

Визначення термінів і призначення документа

Архітектура тут і надалі буде означати формальний структурний опис компонент і регламентів взаємодії з розширеною деталізацією прикладного рівня.

Призначення документа:

- ознайомити виконавця з поточною архітектурою ERP/1;
- визначити технології, які існують й використовуються в ERP/1;
- визначити вимоги та обмеження реалізації.

Рівні архітектури компонентів системи

За основу взята архітектурна методологія ISO/IEC/IEEE 42010 та фреймворк Закмана.

1. Компоненти архітектури ERP/1 є цілком визначеними та узгодженими.
2. Контрактори, які реалізують проєкт нової функціональності ERP/1 повинні зазначити на наступних архітектурних рівнях які модифікації потребує система:
    - Сутності (Resources, ER-діаграми);
    - Форми (Forms, містять додаткову інформацію для UI);
    - Валідації (Validations);
    - Процеси (Послідовність використання API, BPMN діаграми);
    - Довідники (Dictionaries);
    - Права (Scopes);
    - Функції сервісів (публічні та приватні) (Functions, API, RPC);
    - SQL-таблиці;
    - KV-простори;
    - Cache-простори;
    - Правила обмеження швидкості засобами KONG (Rate limiters).
3. Перелік модифікацій повинен надаватися в формі, де кожна зміна зазначена у відповідному документі-вимозі до оформлення кожного архітектурного рівня. В цьому контексті вимоги до опису архітектури є кореневим документом-вимогою який описує зміни до архітектури ERP/1 на верхньому архітектурному рівні.

Маніфест відповідності архітектурі ERP/1

Мотиваційна частина: мета документу.

1. Стандартизація та уніфікація функцій ПЗ повинна бути забезпечена за рахунок використання сучасних інструментальних програмних засобів, які підтримують єдину технологію проектування і розробки функціонального, інформаційного та програмного забезпечення.
2. ПЗ в цілому та інші програмні компоненти повинні відповідати основним міжнародним та національним угодам і стандартам в галузі інформаційних технологій.
3. Необхідно використовувати вже наявні у Замовника системи та програмне забезпечення.
4. Версії систем і підсистем продуктів повинні відповідати версіям, що використовуються.
5. Нове технічне забезпечення (наприклад, підсистеми) має відповідати наступним вимогам:
    - безкоштовність нової системи/підсистеми, її залежностей, і їх ліцензій;
    - наявність підтримки розробника (1 рік від останньої версії);
    - програмне забезпечення з відкритим вихідним кодом (open source).
6. Вимоги стосуються безпосередньо і програмного забезпечення, розробку якого сплачує Замовник.
7. Будь-яке інше програмне забезпечення повинно відповідати вимогам.
8. Вимоги до горизонтального масштабування (за необхідності). Шардінг, контейнеризація, оркестрація.
9. Балансування навантаження між декількома екземплярами сервісів (подами).
10. Використання черг для часомістких операцій.
11. Мінімізація розміру інформаційних повідомлень і оптимізація трафіку.
12. Вимоги до телеметрії, кількість операцій за секунду, латенсі, помилки, статуси.
13. Вимоги до потужності і ємності.
14. Вимоги до логування.
15. Можливість проведення автоматичного тестування з відключеними сервісами КЕП.
16. Мінімізація вартості володіння.

Компоненти системи

Перелік категорій компоненти згідно з класифікацією по їх функціональним категоріям (в дужках зазначені приклади які згодом можливо зафіксуються в технічному завданні).

- Data Storages (Приклади: MongoDB, SQL, S3, Cassandra, Riak, tikv);
- PubSub Buses (Приклади: Kafka, MQTT, AMQP);
- Application Services (Приклади: D3, ЄСІКС, CRM, ЄДРДР);
- Facade Management (Приклади: Traefik / KONG); OSI Layer 4—7;
- Virtual Machines (Приклади: JVM, CLR, BEAM);
- External Systems (Трембіта: ДССУ, Дія, ПФУ, Мінюст).

Види Data Storages:

- KV. Сховище з єдиним простором ключів для зберігання подій і документів.
- SQL. Реляційне сховище для обліку бізнес-сутностей.
- S3. Сховище медіа-об'єктів (документів, файлів, тощо) з Amazon S3 інтерфейсом.
- FS. IPFS, NTFS, SAMBA, GlusterFS, ZFS.

Види Pub/Sub Buses:

- MQTT. Брокер повідомлень, основа Service Oriented Architecture (SOA). ISO.
- AMQP. Теж. ISO. Не використовується зараз.
- XMPP. Теж. ISO.
- Kafka. Non-ISO.
- NSQ. Теж. Non-ISO.

Application Services:

- Anti Fraud. Сервіс блокування зловмисних дій.
- One Time Password. Сервіс верифікації через одноразові паролі.
- Authentication and Authorization. Сервіс аутентифікації і авторизації.
- Integration Layer. Основні сервіси і реєстри.
- Data Layer. Безпосередній доступ до реплік PostgreSQL.
- Digital Signature. Сервіс елекронного підпису.
- Master Citizen Index (MCI). Сервіс ізоляції персональних даних користувачів.
- Extract, Transform, Load (ETL). Зовнішні конектори.

Web Frameworks:

- Основані на шаблонізаторах (PHP, Django DTL, Elixir EEX)
- Основані на DSL (PureScript, Ocsigen, LiveView, N2O, WebSharper, UrWeb, Flutter)
- Клієнтські REST фреймворки з інфраструктурою (vue.js, react.js)
- REST мікро-фреймворки
- WebSocket фреймворки

Вимоги до безпеки

Другий підрозділ описує загальні і технічні вимоги до інформаційної безпеки, що діє наскрізно на прикладному, презентаційному, сесійному і транспортному рівні моделі OSI.

Загальні вимоги до інформаційної безпеки

Додати вимоги для персональних даних, приналежність до кластерів.

1. Вимоги до інформаційної безпеки повинні відповідати:
    - закону України «Про захист інформації в інформаційно-телекомунікаційних системах»;
    - закону України «Про захист персональних даних»;
    - закону України «Про основні засади забезпечення кібербезпеки України»;
    - закону України «Про інформацію»;
    - правилам забезпечення захисту інформації в інформаційних, телекомунікаційних та інформаційно-телекомунікаційних системах, затвердженим Постановою Кабінету Міністрів України від 29.03.06 № 373;
    - порядку функціонування електронної системи охорони здоров'я, затвердженому постановою Кабінету Міністрів України від 25.04.2018 № 411;
    - іншим нормативно-правовим актам, що регулюють питання захисту інформації, у тому числі в інформаційно-телекомунікаційних системах.
2. Створення, адміністрування, розгортання та забезпечення функціонування Системи як засобу інформатизації здійснюється відповідно до вимог Постанови Кабінету Міністрів України від 21.02.2025 № 205 «Деякі питання створення, адміністрування та забезпечення функціонування засобу інформатизації».
3. Виконавець має забезпечити функціональну можливість засвідчення даних, що вносяться до ERP/1 користувачами виключно за допомогою кваліфікованого електронного підпису (далі – КЕП):
    - в якому наявні атрибути `certificate-values` та `revocation-values`, які свідчать про CAdES-X Long формат довгострокового розширеного підпису КЕП;
    - що зберігається на захищеному апаратному носії.
4. Для підключення до ERP/1, виконання функцій формування / перевірки КЕП, для взаємодії з кінцевими користувачами в складі додатка мають використовуватись засоби криптографічного захисту інформації, які мають чинний позитивний експертний висновок за результатами державної експертизи у сфері криптографічного захисту інформації щодо можливості їх використання для криптографічного захисту конфіденційної інформації (персональних даних фізичних осіб).
5. Додаток повинен забезпечувати захист від вразливих компонентів, а саме:
    - усі компоненти, включаючи бібліотеки та плагіни, повинні регулярно перевірятися на наявність відомих вразливостей, зазначених в базі даних exploit-db (`www.exploit-db.com/`);
    - усі застарілі або вразливі компоненти повинні бути оновлені або замінені на безпечні аналоги не пізніше 5 робочих днів з дня виявлення;
    - система повинна бути налаштована таким чином, щоб уникати використання компонентів, які більше не підтримуються.
6. Додаток повинен забезпечувати захист від небезпечних налаштувань, а саме:

- всі непотрібні функції та служби повинні бути вимкнені або видалені;
- конфігураційні файли повинні бути захищені від несанкціонованого доступу і зберігатися в зашифрованому вигляді.

7. Додаток повинен забезпечувати захист від порушень управління доступом, а саме:
   - доступ до всіх ресурсів повинен здійснюватися згідно з принципом найменших привілеїв;
   - контроль доступу повинен бути централізованим і надавати права тільки тим користувачам, які мають на це необхідні підстави;
   - необхідно вести журнал аудиту доступу до конфіденційних даних з можливістю перевірки дій користувачів, термін зберігання яких не менше 3 місяців.
8. Додаток повинен реалізовувати основні функції захисту:
   - забезпечення періодичного резервного копіювання баз даних додатку;
   - забезпечення однозначної ідентифікації та аутентифікації користувачів додатка;
   - забезпечення захисту організаційними заходами від несанкціонованого доступу до інформації при її обробці засобами додатка;
   - контроль за проходженням на мережевому рівні тільки дозволених інформаційних потоків з боку телекомунікаційних мереж до додатку, а також у зворотному напрямку (щонайменше, на рівнях L3, L4 моделі Open Systems Interconnections, OSI);
   - забезпечення реєстрації подій, пов'язаних з отриманням користувачами доступу до ресурсів додатка (проходження/непроходження автентифікації), зміни налаштувань програмного забезпечення, серверів та комутаційного обладнання;
   - забезпечення ведення журналів безпеки, усі важливі події безпеки, такі як спроби несанкціонованого доступу, повинні бути зафіксовані у журналах безпеки. Журнали повинні зберігатися не менше 3 місяців, в захищеному вигляді та бути недоступними для модифікації неавторизованими користувачами;
   - забезпечення конфіденційності та цілісності інформації з обмеженим доступом, що передається каналами зв'язку через незахищене середовище.
9. Заборонено використовувати проміжні інтерфейси авторизації користувачів, крім вебсторінки авторизації.
10. Інформаційна безпека додатка має обов'язково відповідати 10 найбільш важливим перевіркам на безпеку (TOP 10), які розміщуються в інформаційному онлайн-документі для розробників і тестувальників з безпеки додатків The Open Worldwide Application Security Project (OWASP).

Технічні вимоги до інформаційної безпеки

1. Додаток повинен мати відкритим лише порт 443 для безпечного HTTPS-з'єднання. Інші порти повинні бути закриті.
2. Додаток повинен використовувати чинний SSL/TLS сертифікат, що видається лише довіреним органом сертифікації.
3. Додаток повинен використовувати SSL/TLS сертифікат, який відповідає криптографічним стандартам ECC (ECDSA) або RSA (DSA, з 2048-бітними ключами, дозволеними до 2030 року згідно з нормативами NIST).
4. Додаток повинен забезпечувати захист від SQL-ін'єкцій й використовувати екранування спецсимволів: `'` (0027), `"` (0022), `\` (005c), `-` (002d002d), `%` (0025), `/* */` (002f002a, 002a002f), `#` (0023), `&` (0026), `_` (005f), `\n` (005c006e), `\r` (005c0072), `\t` (005c0074), `%00`.
5. Додаток повинен забезпечувати захист від вразливостей автентифікації та керування сесіями, а саме:
   - сесійні токени повинні зберігатися у cookie з прапорами `Secure` та `HttpOnly`;
   - заборона одночасної сесії (сеансу) користувачів на різних пристроях.
6. Додаток повинен забезпечувати захист від XSS-ін'єкцій, а саме:
   - додаток повинен використовувати тільки безпечні заголовки (CSP, HTTPOnly, X-XSS-Protection, X-Frame-Options, HSTS) для запобігання доступу клієнтських сценаріїв до даних;
   - при внесенні користувачем даних в інтерфейсі додаток повинен попередити користувача та не дозволити ввести символи та конструкції (що заповнюються в полях), які можуть бути використані для впровадження скриптів або зміни URL, такі як: `javascript:`, `vbscript:`, `data:`, `onclick`, `onload`, `onsubmit`, `?` (003f), `'` (0027), `"` (0022), `&` (0026), `<` та `>` (003c 003e), `\n` (005c006e), `\r` (005c0072), `\t` (005c0074), `%00`, `/* */` (002f002a, 002a002f), `NULL` та `#` (0023).
7. Додаток повинен забезпечувати захист від виконання довільного коду (Command Execution), наприклад, `eval()`, `exec()`, або `shell_exec()`.
8. Додаток повинен використовувати протокол "Transport Layer Security (далі – TLS)" версії не нижче 1.3, що відповідає вимогам чинного законодавства.

Вимоги до потужності та ємності

Цей підрозділ визначає процес аналізу та валідації вимог до потужності і ємності нової функціональності, що є одними з головних технічних ризиків при проєктуванні інформаційних систем.

1. Програмний код повинен бути сумісний з горизонтальним масштабуванням процесингових обчислень, щоб динамічно збільшувати ємність системи як лінійну залежність від кількості екземплярів процесингових обчислень.
2. Програмний код має автоматично підтримувати одночасне виконання багатьох невеликих процесів (concurrent) та працювати паралельно на кількох ядрах (parallel).
3. У випадку релізу, який додає функціональність і/або змінює flow по роботі з наявним, слід планувати проведення навантажувального тестування з метою визначення впливу на продуктивність системи на середовищі Stage.
4. Програмне забезпечення, що реалізується, повинно відповідати наступним критеріям:
    - Первинне джерело даних, на базі яких проводиться аналіз ризиків потужності та ємності;
    - Якщо аналітичних даних недостатньо, то прогнозування навантаження повинно відбуватися на основі дотичних сервісів;
    - Після прийому релізу відділ контролю якості перевіряє виконання вимог до потужності шляхом навантажувального тестування, який передбачає також взаємодію з відділом супроводу і підтримки.
5. Продуктивність або потужність підсистем визначається наступним набором характеристик (таблиця, яка заповнюється бізнес-аналітиками на основі прогнозованих значень):
    - час реакції (активація асинхронної задачі);
    - пропускна здатність (кількість виконаних задач за одиницю часу);
    - затримка передачі (latency, час приходу першого байту від асинхронної чи синхронної відповіді).

Таблиця характеристик продуктивності

Характеристика	Опис	Прогноз
Час реакції	Активація асинхронної задачі	< 200 мс
Пропускна здатність	Кількість виконаних задач за одиницю часу	500 задач/с
Затримка передачі (latency)	Час приходу першого байту відповіді	< 150 мс

Вимоги до системи логування

Загальні вимоги

1. Змістовність. Кожне повідомлення в логах повинно містити цінну інформацію.
2. Цілісність. Логи повинні утворювати логічний ланцюжок подій, що призвів до проблеми.
3. Відтворення проблем. Логи повинні містити достатньо інформації для відтворення проблеми на середовищі розробника.
4. Роздільність. Кожен сервіс повинен мати власний лог.
5. Лаконічність. Логи повинні бути читабельними та не містити зайвої інформації.
6. Захист даних. Приватні та персональні дані (логіни, паролі, секретні ключі, персональні дані клієнтів) не повинні зберігатися в логах в явному вигляді. Використовуйте хешування.
7. Метрики та статистика. Логи повинні дозволяти побудувати метрики, статистику швидкодії та продуктивності додатка і його компонентів.
8. Історія подій. Логи повинні містити історію подій до моменту виникнення помилки без потреби запитувати користувачів.
9. Логування помилок. Логи помилок повинні бути пов'язані з усіма попередніми повідомленнями, що стосуються процесу або запиту. Повідомлення мають бути доступні для пошуку за унікальним ідентифікатором.
10. Логування змін. Будь-які зміни в конфігурації додатка та його компонентів повинні бути залоговані.
11. Структуровані логи. Логер повинен бути структурованим і логувати інформацію в форматі JSON.

Рівні логування

Рівні логування мають керуватись на рівні конфігурації додатку. Вмикаючи в конфігурації системи один з рівнів логування, також будуть залоговані й повідомлення нижчих рівнів. Наприклад, увімкнувши рівень `information`, ви матимете записи логів `warning`, `error`, `critical`.

1. Trace. Логуються всі властивості об'єктів, параметри методів, виклики методів та виконання кожного рядка коду. Використовується тільки в середовищі розробки/тестування.
2. Debug. Логуються системні параметри, запити до джерел даних, виклики методів. Використовується на продуктивному середовищі на короткий час для діагностики.
3. Information. Логуються події, такі як запуск/зупинка компонентів.
4. Warning. Логуються підозрілі або некритичні позаштатні ситуації.
5. Error. Логуються події, що переривають поточні операції, але не впливають на подальшу роботу системи.
6. Critical. Логуються критичні помилки, що перешкоджають подальшій роботі системи (наразі в системі помилка відсутня).

Інформація для вендорів про інструментарії логування

1. Filebeat:
    - потрібно використовувати для збору логів з усіх джерел;
    - повинен бути налаштований на надсилання даних до Elasticsearch.
2. Logstash:
    - має забезпечувати обробку логів з можливістю трансформації, фільтрації та агрегації даних;
    - конфігурації повинні бути задокументовані та доступні.
3. Elasticsearch:
    - логи повинні зберігатися в Elasticsearch для швидкого пошуку та доступу;
    - необхідно забезпечити належну масштабованість та продуктивність системи.
4. Kibana:
    - має використовуватися для візуалізації та аналізу збережених логів;
    - потрібно забезпечити можливість створення графіків та дашбордів для моніторингу.

Події логування

1. Запуск/зупинка окремих сервісів системи.
2. Події безпеки типу звернення системи до сервісів системи.
3. Помилок у роботі системи, таких як комунікаційні, цілісності даних у системі, непередбачувані затримки в обробці інформації.
4. Критичні події від системи моніторингу (критичний обсяг пам'яті, дискового простору тощо).
5. Інші події безпеки.

Структура журналу логування

Структура журналу логування повинна включати:

1. Ідентифікатор запиту в наборах відкритих даних, на який надається відповідь.
2. Дата та час запиту.
3. Результат обробки запиту.
4. Ідентифікатор відповіді від набору відкритих даних, щодо якого надається повідомлення.
5. Дата та час створення відповіді.
6. Набір вхідних параметрів.
7. Час виконання.

Вимоги до структурних логів ELK стеку

Лог має бути представлений у форматі JSON об'єкта з наступними визначеними полями:

- `time` — log time
- `severity` — log level (`DEBUG`, `INFO`, `WARNING`, `ERROR`)
- `log` — log message

Додатково включати по наявності наступні поля, якщо присутні такі метадані:

- `source_location.file`
- `source_location.line`
- `source_location.module`
- `source_location.function`
- `error.initial_call` — `"#{module}.#{function}"` або `"#{module}.#{function}/#{arity}"`
- `error.reason` — `"#{module}.#{function}"` або `"#{module}.#{function}/#{arity}"`

Додатково можна включати інші поля за необхідності.

Вимоги до контролю якості

Загальні вимоги до обліку і контролю якості

Види тестувань:

- Внутрішні тести розробників.
- Ручне тестування за допомогою Postman, curl, wscat, тощо.
- Автоматичне регресивне тестування з використанням кваліфікаційного електронного підпису.
- Навантажувальне тестування.

Тестувальна документація має трактувати кроки, терміни та результати тестування. Склад документації з тестування залежить від конкретної задачі, але повинен містити:

- Об'єкт тестування. Необхідно чітко визначити, який об'єкт тестування підлягає приймальному тестуванню (модуль, програмне забезпечення, веб-додаток, мобільний додаток або комплексна система). Вказати конкретну функціональність та компоненти.
- Критерії успішності. Критерії, які повинні бути виконані для визнання тестування успішним (функціональні вимоги, продуктивність, стабільність, безпека, сумісність тощо).
- Сценарій тестування. Опис тестового середовища, набір сценаріїв тестування та покрокових процедур (тест-кейси).
- Умови тестування. Конфігурації системи, вхідні дані, налаштування, середовища виконання тощо.
- План тестування. Документ, що описує весь обсяг робіт із тестування.
- Критерії прийняття. Кількість та тип прийнятних дефектів, виконання функціональностей, бізнес-сценаріїв тощо.
- Протоколи результатів тестування. Загальна оцінка (успішно/неуспішно), вплив тестового середовища, результати кожного тест-кейсу.
- Стратегія контролю версій. Контроль та управління версіями документів, які використовуються в тестуванні.
- Вимоги до навантаження та продуктивності. Обсяги даних, кількість одночасних користувачів, транзакцій, очікувані показники продуктивності та час відповіді.
- Звітність та документування. Формат звітів, структура, висновки та рекомендації.

Вимоги до ручного тестування

Ручне тестування повинно оцінювати функціональність з точки зору звичайного користувача та надавати максимально розгорнутий і зрозумілий звіт. Необхідно виконати перевірку:

- форм користувача (коректне заповнення, виведення помилок, обов'язкові/необов'язкові поля, календарі, меню тощо);
- посилань та навігації сайту;
- форм реєстрації та авторизації;
- роботи з базою даних (додавання, редагування, видалення даних, завантаження файлів);
- файлів cookie;
- пошукового рядка.

Необхідно виконати перевірку нетипових сценаріїв для виявлення несподіваних помилок і багів.

Перевірка безпеки:

- Конфіденційність (обмеження доступу до особистих даних);
- Цілісність (можливість відновлення інформації після атаки);
- Доступність (розмежування рівнів доступу).

Оцінка дизайну:

- Простота експлуатації;
- Зручність навігації;
- Відповідність контенту (шрифти, кольори, розташування елементів);
- Мінімальна кількість кліків для виконання операцій.

Тестування на сумісність:

- На різних пристроях (ПК, планшет, телефон) та ОС (Windows, macOS, Android);
- У різних браузерах (Edge, Firefox, Chrome, Safari, Opera).

Тестування продуктивності

- Навантажувальне тестування;
- Стрес-тестування;
- Об'ємне тестування;
- Тестування надійності.

Необхідно виконати регресивне тестування після внесення змін у код. Функціональні вимоги до HTTP API тестування:

- Правильність відповідей та використання коректних HTTP-статус-кодів
- Валідація вхідних даних (формати, обов'язкові поля, медичні коди)
- Захист від помилок та чіткі повідомлення про помилки
- Правильний формат даних у відповідях
- Перевірка обов'язкових полів.

Вимоги до специфікацій технічної документації: Технічна документація повинна містити:

- Архітектурний опис системи та взаємодії компонентів

- Опис модулів та функцій
- API-специфікацію (ендпоінти, методи, запити, відповіді, статуси)
- Опис бази даних (схема, таблиці, зв'язки);
- Опис інтерфейсу користувача (wireframes, прототипи, адаптивність)

Вимоги до формату сценаріїв та протоколів тестування

Сценарії надаються у вигляді таблиці У полях «Від замовника» та «Від виконавця» вказується ПІБ. Зазначаються дати надання та виконання сценаріїв. Протокол тестування формується у вигляді звіту Jira або у форматі XLS/DOCX. Протокол повинен містити:

- Найменування системи та номер версії;
- Назву тестового середовища;
- Перелік виконаних тест-кейсів.

Для кожного тест-кейсу вказується:

- Назва тест-кейсу;
- Статус;
- Дата виконання;
- ПІБ тестувальника;
- Перелік незакритих дефектів;
- Підсумковий висновок.

Табл. 3: Шаблон протоколу тестування

ID сценарію	
Назва сценарію	
ID тестового випадку	
Передумови	
Кроки відтворення / Умови	
Очікуваний результат	
Статус	
Коментар	
Дата тестування	
Тестувальник	
Від замовника	
Від виконавця	

Вимоги до плану тестування

Перед початком тестування функціональності має бути підготовлено план тестування. План тестування — це документ, в якому визначені об'єми, ресурси та календарний план тестування. Визначається відділом контролю якості. План тестування повинен містити наступну інформацію:

- Об'єкт тестування. Назва проекту або компонента, який підлягає тестуванню.
- Мету тестування. Виявлення дефектів, перевірка функціональності, валідація вимог, оцінка продуктивності тощо.
- Стратегія тестування. Типи тестів: модульне, інтеграційне, системне, регресійне, продуктивності, безпеки тощо.
- Обсяг тестування. Кількість тестових сценаріїв, наборів даних, ітерацій, тривалість тестового циклу.
- Ресурси тестування. Людські ресурси, апаратне та програмне забезпечення, тестові середовища, тестові дані.
- Графік тестування. Дати початку та закінчення тестового циклу, терміни проведення окремих видів тестів.
- Ризики тестування. Ідентифікація, оцінка ризиків та заходи щодо їх зниження.
- Критерії прийняття. Умови прийняття системи після тестування.
- Відповідальності та ролі. Ролі та відповідальності учасників тестування (тестувальники, керівники проекту, розробники тощо).

Вимоги до автоматичного тестування

Вся функціональність сервісу повинна бути повністю покрита автоматичними тестами.

Виконавець зобов'язаний надати документацію, яка містить:

- Об'єкт автотестування;
- Мету автотестування;
- Типи тестів та сценарії;
- Використовувані фреймворки та інструменти;
- Критерії успішності;
- Інтеграцію в процес CI/CD;
- Описи тестів, підготовку тестових даних, передумови та очікувані результати.

Вимоги до результатів приймального (UAT) тестування

Приймальне (UAT) тестування має бути виконано на Demo-середовищі (якщо інше не погоджено з Замовником). Замовник затверджує висновок про відповідність функціональності вимогам технічного завдання згідно з таблицею 5.5.1. Після успішного UAT Виконавець передає поставку програмного забезпечення для інсталяції та налаштування на середовищах Замовника. Прийняття функціональності можливе повністю або частинами згідно з календарним планом. Функціональність має бути прийнята Замовником в повному обсязі згідно з календарним планом виконання робіт.

Вимоги до інтерфейсів

Даний підрозділ визначає ергономічні та технічні вимоги до інтерфейсів користувача адміністративних панелей.

Загальні вимоги

Інтерфейс користувача адміністративної панелі є критичним елементом для ефективного управління та адміністрування програмного забезпечення чи веб-сайту.

- Інтуїтивність та простота. Інтерфейс повинен бути легким для використання, навіть для користувачів без глибоких технічних знань.
- Логічна структура та навігація. Забезпечення логічної організації елементів та легкої навігації між різними функціями та розділами.
- Мобільна сумісність. Адаптивний дизайн для коректного відображення на різних пристроях, включаючи мобільні телефони та планшети.
- Гнучкість налаштувань. Можливість налаштування інтерфейсу відповідно до унікальних потреб та завдань адміністратора.
- Зручне управління даними. Легкість управління, внесення та вилучення даних з адміністративної панелі.
- Інформативність. Забезпечення користувача інформацією про стан системи, статистику та повідомлення про можливі проблеми.
- Безпека та авторизація. Захист інтерфейсу від несанкціонованого доступу, використання протоколів шифрування та можливість налаштування рівнів доступу.
- Інтеграція з іншими системами. Можливість інтеграції з іншими системами та сервісами.
- Спрощений процес розгортання та оновлення. Легкість встановлення та оновлення адмінпанелі для мінімізації перерв у роботі.
- Підтримка різних мов. Locaize (продукт який використовується в кабінеті пацієнта).
- Підтримка технічних засобів. Забезпечення сумісності з різними веб-браузерами та операційними системами.

Вимоги до відображення інформації

- Кросбраузерність. Система має бути адаптованою для використання в усіх сучасних браузерах.
- Адаптація для людей з вадами зору. Інтерфейс має бути адаптований для людей з порушенням зору згідно з ДСТУ EN 301 549:2022 та рекомендацій WCAG 2.1 (W3C Web Content Accessibility Guidelines):
  - контрастні кольори;
  - вибір розміру шрифту.
- Адаптація до розмірів екрана на пристроях. Веб-сайт має бути адаптивним і коректно масштабуватися на екранах різного розміру.

Вимоги до швидкодії додатка

Швидкість завантаження та відображення табличних частин. Час повного завантаження таблиці з 2000 рядками повинен становити не більше 1 с, а відображення видимої частини повинна становити не більше 300 мс.

Вимоги до технології реалізації додатку

- Single Page Application. Використання наперед сформованих HTML-частин на стороні сервера і модифікація структури DOM-елементів в браузері відносно WebSocket.
- Data-on-Wire. Використання тільки даних між клієнтом і сервером, що вимагають розвиненої інфраструктури (шаблонізаторів і клієнтської шини повідомлень для графічних контрольних елементів) і мов програмування для об'ємних клієнтів.
- Figma. Артефакти додатку (assets) та/або демо-прототип UI/UX повинні надаватися в Figma.

⁎

Висновок

Дотримання принципу «політики → вимоги → технічне завдання» та обов'язкове використання ДСТУ гарантує технологічну незалежність, аудитованість та відповідність державним нормам. Конкретні продукти обираються лише на етапі технічного завдання.

⁎

Література

[1] Міністерство охорони здоров'я України. Технічні вимоги Реабілітація 2.0. Версія 08.08.2024.

[2] ДСТУ 3973-2000. Система розроблення та постачання продукції на виробництво. Правила виконання науково-дослідних робіт. Загальні положення.

[3] ДСТУ 3974-2000. Система розроблення та постачання продукції на виробництво. Правила виконання науково-конструкторських робіт. Загальні положення.

[4] ДСТУ 42010:2018. Інженерія систем і програмних засобів. Опис архітектури.

[5] Закон України «Про захист інформації в інформаційно-телекомунікаційних системах».

[6] Закон України «Про захист персональних даних».

[7] Закон України «Про основні засади забезпечення кібербезпеки України».

[8] OWASP Foundation. OWASP Top 10: The Ten Most Critical Web Application Security Risks. `https://owasp.org/www-project-top-ten/`.

[9] NIST Special Publication 800-57 Part 1 Revision 5. Recommendation for Key Management.

[10] ISO/IEC/IEEE 42010:2022. Systems and software engineering — Architecture description.

[11] Zachman, J.A. A Framework for Information Systems Architecture. IBM Systems Journal, 1987.