

А.Д. Тарасенко, А.Ю. Дорошенко

ВЕБ-СЕРВІС ДЛЯ СИСТЕМ УПРАВЛІННЯ РЕСУРСАМИ ПІДПРИЄМСТВА

Розроблено веб-сервіс для систем управління ресурсами підприємства, що дозволяє створювати та управляти швидкими захищеними системами управління ресурсами підприємства на основі необхідних користувачеві модулів. Унікальність сервісу полягає в тому, що окрім наявності більшості вже існуючих стандартів та можливостей інших ERP систем від інших ІТ-компаній, він має суттєву перевагу – в його використанні відсутня необхідність звертання до ІТ-спеціалістів задля розробки та налаштування ERP системи під потреби кінцевого користувача. Здійснено розробку головного веб-сайту веб-сервісу, який передбачає автентифікацію та авторизацію; додатку інсталятора для створення і запуску ERP системи; додаток серверу ERP системи, та клієнтського додатку для взаємодії із сервером ERP системи. Передбачено створення облікових записів працівників компанії у ERP системі, їх розподіл за відділами для доступу до встановлених модулів, а також створення системи як у локальному дата-центрі, так і у хмарному середовищі. Проведено функціональне тестування розробленого веб-сервісу через тестування кожного додатку окремо та у спільному взаємозв'язку. Ключові слова: система управління ресурсами підприємства, ERP система, модульність, мережева взаємодія.

Вступ

Процес організації бізнесу все більше компаній та корпорацій починають із встановлення системи управління її ресурсами та загальним функціоналом автоматизації бізнес-процесів усередині компанії. Відповідно, для вирішення даного питання бізнес звертається до послуг ІТ-спеціалістів для створення даної системи. В більшості випадків ці системи мали схожий функціонал, схожі сервіси й можливості, тому із часом у деяких провідних ІТ-компаній виникла ідея написання даної системи із усіма основними функціями та можливостями та постачання її як ІТ-послуги. Але оскільки ніколи не відомо, який функціонал буде необхідним потенційному клієнтові, від процесу створення програмних доповнень до вже існуючої системи постачальники даного сервісу вирішили не відмовлятися. В результаті маємо ERP систему із базовим функціоналом, яка за необхідності доповнюється потрібними клієнтові механізмами, та постачається даному клієнтові у вигляді ІТ-послуги. Клієнтові це вигідно, бо йому не потрібно витратити час та ресурси (в основному грошові – на наймання ІТ-спеціалістів на розробку системи), а постачальнику системи дає прибуток за користування цією системою клієнтом.

У даній роботі натомість реалізовано систему, призначену для використання як ІТ-послуги на потенційних комерційних засадах. Кінцевими користувачами даної системи є адміністратори або ІТ-спеціалісти компаній, яким поставлена задача впровадження ERP системи у даній компанії.

1. Стратегії та основні компоненти

ERP система – інформаційна система управління ресурсами підприємства, належить до програмного забезпечення, що її компанії, корпорації та організації використовують для управління повсякденними процесами всередині своєї бізнес-структури, такими як бухгалтерський облік, управління проектами, закупівлею, операціями з ланцюгами поставок тощо. «Під інформаційною розуміють будь-яку систему, яка за допомогою технічних засобів виконує одну, або кілька функцій, як збирання, передавання, перетворення, накопичення, зберігання, та обробка інформації» [1-2].

Повний пакет ERP системи також включає управління ефективністю діяльності підприємства, а також програмне забезпечення, яке полегшує процеси планування, складання бюджету, прогнозу-

вання та звітності про фінансові досягнення компанії [3].

До основних характеристик стратегії ERP належать:

- централізація: використання єдиної системи транзакцій для переважної більшості операцій та бізнес-процесів підприємства. Усі операції об'єднані в єдину базу даних для подальшої обробки та збалансованих планів у реальному часі;

- відтворюваність: забезпечення можливості використання одного пакету програмного забезпечення для різних організацій (можливо, з різними налаштуваннями та розширеннями);

- локалізація: підтримка в єдиній системі декількох валют та мов;

- підтримка декількох юридичних осіб, кількох підприємств, декількох облікових політик та різних схем оподаткування в єдиній системі; це необхідно для використання у корпораціях, зокрема, транснаціональних.

Окрім цього, є ключові компоненти сучасної ERP-системи, які мають вирішити головні проблеми більшості компаній у розгляді питання про перехід до хмари. Є сім компонентів, які поділяються на дві категорії – сучасні параметри платформи та сучасний дизайн бізнес-додатків. Разом ці сім компонентів визначають стандарти сучасної ERP-системи. До того ж, сучасні хмарні додатки мають масштабуватися відповідно до вимог бізнесу та підтримувати новітні цифрові технології для задоволення потреб організації [4].

До вищезгаданих компонентів сучасної ERP-системи належать:

- безпека. Багатошаровий підхід до захисту даних на кожному шарі стеку є головним для максимального захисту даних [5]. Використання безпечної архітектури ізоляції даних у хмарі зменшує ризик та забезпечує швидший доступ до даних та їх обробку;

- інтеграція. Хмарні рішення ERP мають безперешкодно поєднувати бізнес, людей та процеси. Також має бути продумана та реалізована можливість підключення до інших хмар, локальних систем та сторонніх рішень. Для забез-

печення сумісності та масштабованості слід вибрати рішення, яке використовує загальну структуру, засновану на галузевих стандартах;

- персоналізація. Хмарні рішення, побудовані на платформі, що базується на стандартах, пропонують персоналізацію та конфігурацію в додатку, що призводить до покращень, безпечних для оновлення. Якщо локальні налаштування клієнта потрапляють у такі сфери, як робочі процеси, інтеграція та звітування, є велика ймовірність вирішення вимог бізнесу за допомогою хмарних рішень;

- повнота. Вбудовані найкращі практики підтримують стандартизацію, що дозволяє знизити витрати та збільшити продуктивність. Навіть якщо хмарний перехід поступовий, доступ до повного набору інтегрованих передових бізнес-процесів забезпечує стандартизацію підприємств;

- глобалізація. Вихід на нові ринки та географії створюють складність, вимагаючи будь-якого хмарного рішення ERP для підтримки декількох дочірніх компаній та локалізації країн. Часто місцеві центри обробки даних повинні відповідати вимогам щодо збереження даних. Правильний хмарний ERP має забезпечувати безперебійний обмін корпоративною інформацією між операціями, бізнес-підрозділами та штаб-квартирою;

- аналіз, заснований на аналізі. Хмарне ERP-рішення повинно мати надійний доступ до даних у реальному часі у своїй основі для забезпечення єдиного джерела істинності між ролями, звітами та аналізом. Це забезпечує своєчасну доставку точних показників ефективності для передових менеджерів та суттєво спрощує процес у разі його залежності від періоду закриття або окремого витягу сховища даних;

- цифрові можливості. Для сучасного бек-офісу цифрові технології мають бути інтегровані в бізнес-процеси та операції для створення цілісного, продуктивного та інтуїтивно зрозумілого досвіду. Користувацький досвід повинен бути цікавим і включати мобільну доступність для постійних співробітників, власну соціальну інтеграцію для безпечної спіль-

ної роботи в контексті та нестандартне оптичне розпізнавання символів для зображення рахунків-фактур.

2. Огляд аналогів

Одним із найбільш популярних та передових сервісів ERP систем є розробка від технологічного гіганта Microsoft – MS Dynamics 365 BC. Це програмне рішення управління для малого та середнього бізнесу з метою автоматизації бізнес процесів. Даний продукт є частиною сімейства Microsoft Dynamics [6]. Насамперед цей продукт є веб-орієнтованим SaaS (англ. «Software as a Service» – «програмне забезпечення як сервіс») рішенням. До версії #14 була доступна локальна версія, у складі налаштувань якої була можливість використання спеціально налаштованого клієнта. Починаючи із версії #15, доступ до системи здійснюється лише через Інтернет на веб сайті dynamics.com.

Іншим не менш популярним та не менш передовим сервісом ERP систем відносно іншого технологічного гіганта – компанії Oracle – є система Oracle ERP [7]. Це інтегрований хмарний комплекс додатків для управління фінансами, закупівлею і портфелем проєктів для малого та середнього бізнесу. Даний продукт побудований на міцному об'єднанні платформи, додатків та підходу з використанням фінансових, закупівельних, портфельних проєктів та управління ефективністю діяльності підприємств.

На відміну від MS Dynamics 365 BC, дана система доступна лише у вигляді хмарного сервісу, без можливості локального встановлення або гібридного розгортання.

В обох даних аналогах присутній суттєвий недолік – кінцевий користувач має звертатися до постачальника даної системи (Microsoft, Oracle тощо) задля налаштування системи під свої потреби. Розроблена система даного недоліка позбувається.

3. Розробка API

API сервер даної системи – її головний компонент. Взаємодія із зовнішніми запитами у даному проєкті реалізована

у вигляді API контролерів – набору методів, які приймають HTTP GET, POST, PUT і DELETE запити, та виконують певну бізнес-логіку.

У даному API присутні три контролери:

- Modules: даний контролер відповідає за роботу із репозиторієм модулів із БД. В ньому присутні наступні методи:

1) GET api/modules?moduleID= повертає детальну інформацію про модуль за його ідентифікатором;

2) POST api/modules/all = повертає масив усіх модулів із врахуванням фільтру (фільтр передається через Body у запиті);

3) POST api/modules/non-default = повертає масив усіх модулів не за замовчуванням із врахуванням фільтру (фільтр передається через Body у запиті);

4) POST api/modules/marketplace = повертає масив усіх невстановлених на вказаній ERP системі (через Body у запиті передається ідентифікатор системи та фільтр);

5) POST api/modules/marketplace/add = запит на додавання модулю до вказаної ERP системи (через Body у запиті передаються ідентифікатори системи та модулю);

6) POST api/modules/marketplace/remove = запит на видалення модулю із вказаної ERP системи (через Body у запиті передаються ідентифікатори системи та модулю);

- ErpSystems: основний контролер у даному додатку. Відповідає за роботу із репозиторієм ERP систем із базою даних. Даний контролер захищений авторизацією (для виклику будь-якого методу необхідно спочатку успішно авторизуватися у системі) через токени, час дії яких – 20 хвилин.

У ньому присутні наступні методи:

1) GET api/erp-systems/check-company?company= перевіряє наявність у головній базі даних сервісу ERP системи із вказаною назвою компанії;

2) GET api/erp-systems/installed-modules?erpID= повертає масив встановлених у вказаній ERP системі модулів за ідентифікатором ERP системи;

3) POST `api/erp-systems/user-erp-systems?userID=` повертає відфільтрований масив ERP систем, які належать вказаному користувачеві за ідентифікатором користувача (фільтр передається через Body у запиті);

4) POST `api/erp-systems` виконує запит на створення ERP системи користувачем;

5) DELETE `api/erp-systems/erp-systemID=` виконує запит на видалення ERP системи користувачем за ідентифікатором ERP системи.

- Users – даний контролер відповідає за роботу із репозиторієм користувачів у базі даних. Аналогічно контролеру «ErpSystems», даний контролер також захищений авторизацією. Володіє наступними методами:

1) POST `api/users/login` виконує запит на авторизацію користувача. Модель на авторизацію передається через Body запиту;

2) POST `api/users` виконує запит на реєстрацію користувача. Модель на реєстрацію передається через Body запиту;

3) GET `api/users/confirm-email?userID=` виконує запит на підтвердження електронної пошти за ідентифікатором користувача;

4) GET `api/users/forgot-password?email=` виконує запит на надіслання листа для відновлення паролю за вказаною електронною поштою;

5) POST `api/users/restore-password` виконує запит на відновлення паролю. Модель на відновлення паролю передається через Body запиту;

6) POST `api/users/change-password` виконує запит на зміну паролю. Модель на зміну паролю передається через Body запиту;

7) PUT `api/users` виконує запит на оновлення даних про користувача (електронна пошта, логін тощо). Модель із новими даними передається через Body запиту;

8) DELETE `api/users` виконує запит на видалення облікового запису користувача. Модель із користувачем передається через Body запиту.

4. Розробка головного веб-сайту

Головний веб-додаток повинен слугувати відразу як ознайомчий веб-сайт, на якому користувач може ознайомитися із усіма послугами даного сервісу (модулями (їхніми описами та деталями) та серверним і клієнтським додатками), так і функціональний веб-застосунок.

Вибираючи потрібну сторінку, користувач може знайти необхідну інформацію про модулі, серверний додаток ERP системи та клієнтський додаток.

Для початку було розроблено 3 модулі: модуль управління (встановлений у системі за замовчування та надає доступ до таблиць працівників компанії, відділів та власне модулів), модуль фінансів (надає базовий функціонал управління бухгалтерським обліком) та модуль логістики (надає базовий функціонал для управління оборотом товарів, транспортування тощо).

5. Розробка інсталлятора

Інсталлятор – головний інструмент користувача для створення ERP системи. Головна мета даного застосунку – дати користувачеві механізм попередніх налаштувань серверу перед його початком роботи.

Важливо зазначити, що перед встановленням системи, необхідно налаштувати правила фаєрволу, щоб зробити комп'ютер, на якому буде встановлений сервер, доступним за його публічною IP адресою.

Задля забезпечення коректної роботи реєстрації систем, даний додаток взаємодіє із головним API сервером. Це відбувається на трьох етапах:

- Авторизація – для запуску системи користувач повинен мати обліковий запис у системі;

- Перевірка наявності системи за назвою компанії – в процесі успішного створення системи їй буде наданий домен. Цей домен генерується залежно саме від назви компанії власника системи за формулою 1:

Домен = «erp-»+назва_компанії+
«.com» (1)

Даний домен слугуватиме ключем шифрування адреси серверу системи (його публічного IP та порту), результатом шифрування буде ключ організації, який надалі використовуватиметься працівниками для успішної авторизації у системі через клієнтський додаток;

- Реєстрація – власне процес відправлення запиту на API для реєстрації нової системи.

6. Розробка серверу ERP системи

Головний компонент – сервер ERP системи – є, по суті, веб-додатком, призначеним для взаємозв'язку між робітниками компанії та базою даних. Мережева взаємодія буде досягнута шляхом використання API контролеру серверу ERP системи.

Сам додаток працює в об'єднанні із інсталятором. Після успішної реєстрації системи через інсталятор генерується JSON файл із налаштуваннями системи. Даний JSON передається у виконуючий файл серверу, де на основі даних налаштувань і конфігурується система. До даних налаштувань належать:

- публічна та приватна IP адреси;
- порт самого додатку, за яким у веб-браузері можна відкрити веб-інтерфейс серверу;
- ключі організації та шифрування;
- домен та ідентифікатор ERP системи;
- перелік модулів;
- шлях підключення до бази даних;
- назва компанії.

Додаток складається із набору сервісів, кожен із яких виконує певну задачу, та головного сервісу, котрий вирішує, який сервіс (або сервіси) необхідно використати залежно від запиту користувача.

Веб-інтерфейс серверу є його візуальною частиною. Він створений для полегшення роботи адміністратора системи і надає весь необхідний інструментарій.

Єдиний спосіб для працівників отримати доступ до функціоналу модулів - це бути частиною відділу, до якого прикріплені необхідні модулі. По суті, відділ є логічним об'єднанням працівників із метою надання їм певних модулів. Тому

для полегшення роботи адміністраторів під час запуску серверу буде автоматично створений відділ із доступом до всіх встановлених модулів. Для початку найму нових працівників достатньо створити відділ із доступом до модулю управління (який, як вже було зазначено, встановлено в системі за замовчуванням), та обліковий запис головного HR (англ. «Human Resources») менеджера і закріпити його за даним відділом.

Створюючи обліковий запис нового працівника, необхідно вказати адресу його електронної пошти, на яку надійде лист із його логіном, паролем (котрий генерується системою закрито від адміністратора та буде відомий лише самому працівнику в листі) та ключем організації. Логін працівника генерується за формулою 2:

$$\text{Логін} = \text{нікнейм_пошти} + \text{«@»} + \text{домен_системи}, \quad (2)$$

де нікнейм_пошти це набір символів із електронної пошти працівника до символу «@» виключно.

Використовуючи дані із електронного листа та клієнтський додаток, працівник компанії може успішно авторизуватися у системі та починаючи роботу.

Аналогічно користувач отримуватиме листи при оновленні або видаленні його облікових записів.

Головний ERP сервіс є основним компонентом, котрий має доступ до усіх інших сервісів та здійснює на них керуючий вплив. Саме даний сервіс обробляє запит користувача та виконує певні дії для виконання необхідної операції.

Не менш ваговим є сервіс взаємодії із базою даних, оскільки саме через нього відбувається взаємодія із базою даних ERP системи. Алгоритм дій залежить від операцій у БД, які потребує користувач. Таких операцій може бути сім:

- GetById – пошук за ідентифікатором;
- Select – повна вибірка із таблиці;
- Insert – запис до таблиці;
- Update – оновлення запису у таблиці;
- Delete – видалення із таблиці;

- SelfUpdate – аналогічно оновлення у таблиці, проте дана операція викликається лише у разі оновлення користувачем власного профілю у клієнтському додатку;

- CreateDB – створення БД. Дана операція викликається лише один раз – на самому початку роботи сервера для створення усіх таблиць на основі встановлених модулів.

Сервіс зовнішньої інтеграції є обов'язковим згідно з технічними та функціональними умовами, та слугує основною ланкою у комунікації між працівниками та системою.

Даний інструмент реалізований у вигляді API контролеру, який має один єдиний HTTP POST метод. Даний метод приймає модель запиту у Body HTTP запиту. Варто зазначити, що відповідно до технічних вимог, запити не передаються мережею в чистому вигляді. Тому на даний сервіс надходять лише зашифровані повідомлення. Після цього, згідно з блок-схемами алгоритму дешифрування повідомлення, відбувається процес декодування (рисунок 1):

Блок 1 – прийом закодованого повідомлення;

Блок 2 – Перевірка, чи починається повідомлення на спеціальний роздільник. Якщо так, то перейти до блоку 4, інакше – перейти до блоку 3;

Блок 3 – Дешифрування повідомлення доменом системи;

Блок 4 – Видалити роздільник із початку повідомлення;

Блок 5 – Дешифрування повідомлення головним секретним ключем;

Блок 6 – Перевірка успішності дешифрування повідомлення. Дана перевірка спрацьовує, якщо під час декодування не виникає жодного виключення (Exception). Якщо ні, то у моделі відповіді позначається, що вона не була успішна та вказується повідомлення про помилку, інакше – перейти до блоку 7;

Блок 7 – При успішному дешифруванні результатом виходить модель запиту у форматі JSON, тому для подальшої роботи необхідно конвертувати даний формат у об'єкт класу.

Після цього об'єкт запиту передається на головний ERP сервіс, де виконується уся необхідна бізнес-логіка. Після цього модель відповіді шифрується згідно алгоритму кодування (рисунок 2):

Блок 1 – надходить модель відповіді;

Блок 2 – модель відповіді серіалізується у формат JSON;

Блок 3 – перевірка операції запиту. Якщо відбувалась операція авторизації у системі (Login), перейти до блоку 5, інакше – перейти до блоку 4;

Блок 4 – зашифрувати JSON головним шифрувальним ключем;

Блок 5 – зашифрувати JSON доменом системи;

Блок 6 – дописати на початок повідомлення роздільник.

У кінці процедури зашифрована відповідь повертається користувачеві.

Задля коректної роботи додатку був використаний стандартний механізм проектування додатків Dependency Injection. Для цього під час запуску додатку генерується колекція абстракцій сервісів, які будуть використані у роботі серверу та їх відповідних реалізацій. Приклад такого налаштування наведено далі.

```
public void ConfigureServices
(IServiceCollection services)
{
    services.AddRazorPages();
    services.AddServerSideBlazor();
    services.AddHttpClient(); services.
    AddHttpContextAccessor(); var
    json = Configuration.GetValue
    <string>(ConfigurationMak
    erService.ServerApplication);
    _erpSystem = Converter.
    FromJson<ErpSystem>(json);
    services.AddSingleton(_erpSystem);
    services.AddTransient<Core.
    Domain.Configuration.
    IConfigurationService,
    ConfigurationService>(); services.
    AddTransient<IServerConfig
    urationService,Configuration
    Service>(); services.AddTran
    sient<IRequestSenderService,
    RequestSenderService>();
```

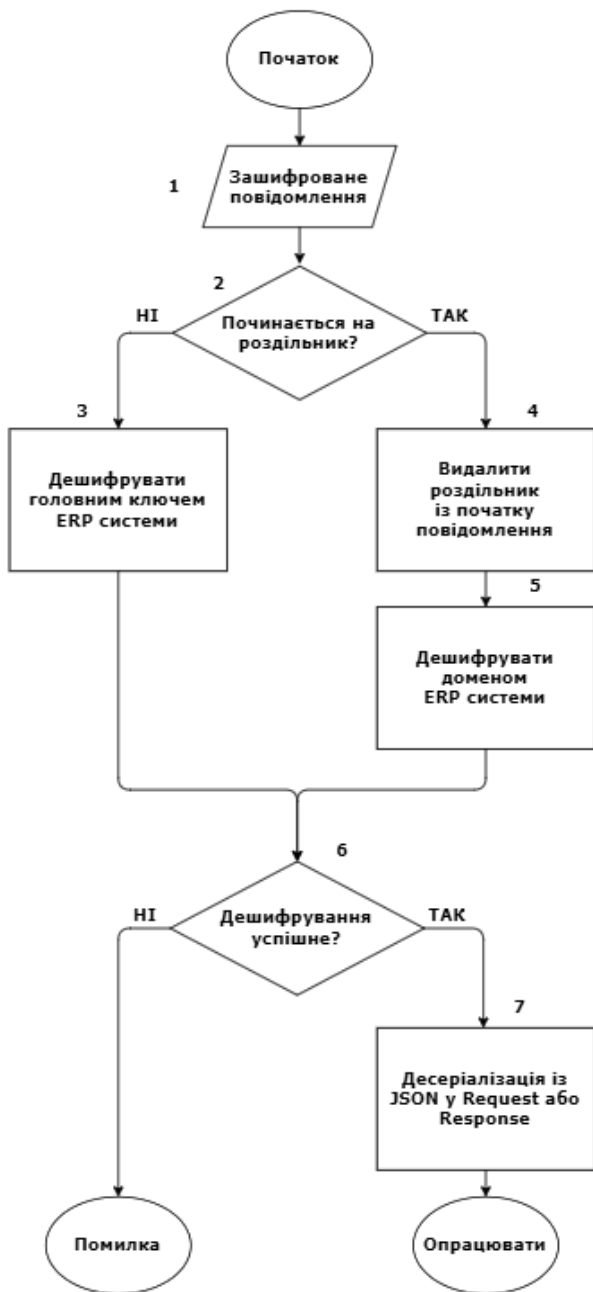


Рис.1 – Блок-схема алгоритму декодування



Рис.2 – Блок-схема алгоритму кодування

```

switch (_erpSystem.DatabaseType)
{
    case DatabaseType.SQL:
        services.AddTransient<IDatabaseService, RelationalDatabaseService>();
        break;

    case DatabaseType.NoSQL:
        switch(_erpSystem.DatabaseProvider)
        {
            case DatabaseProvider.
    
```

```

DynamoDb:
services.
AddTransient<IDatabaseService,
DynamoDatabaseService>();
break;
case DatabaseProvider.MongoDb:
case DatabaseProvider.
DocumentDb:
MongoDatabaseService.
RegisterGuidSerializer();
services.
AddTransient<IDatabaseService,
MongoDatabaseService>();
    
```

```
        break;
    }
    break;
}
services.AddTransient<IEmailSenderService, EmailSenderService>();
services.
AddTransient<ISystemTerminator,
SystemTerminator>();
services.AddTransient<ISettingsSaverService, SettingsSaverService>();
services.
AddTransient<IMainErpService,
MainErpService>();
services.AddTransient<IErpServiceEncryption, ErpServiceEncryption>();
services.
AddTransient<IScreenService,
ScreenService>();
services.AddTransient<ILocalizationService, LocalizationService>();
services.
AddScoped<ICookieService,
CookieService>();
services.
AddSingleton<IStateService,
ServerStateService>();
services.
AddSingleton<IEventBusService,
EventBusService>();
}
```

7. Розробка клієнтського додатку ERP системи

Працівники компанії також повинні мати можливість працювати із ERP системою своєї компанії. Даний додаток є програмними засобом, який надає цю можливість. Цей застосунок є кросплатформним, що робить можливим запускати його на будь-якій операційній системі (Windows, Linux або MacOS).

Аналогічно до серверу ERP системи, даний додаток також поділений на сервіси. Всього їх два:

- сервіс зовнішньої інтеграції: аналогічно до серверу, це API сервіс, який протоколом HTTP та аналогічними алгоритмами шифрування та дешифрування надсилає на сервер системи запити та отримує відповіді;

- головний сервіс, який по суті є обгорткою API сервісу, з метою приховати деталі реалізації зовнішньої взаємодії та не працювати безпосередньо із API сервісом, що полегшить тестування та дасть можливість розширювати його додатковим функціоналом.

Стартовим вікном у даному додатку є сторінка авторизації. Під час авторизації користувач має ввести свої логін, пароль та ключ організації (усі ці дані будуть відомі у електронному листі, який йому надійде після створення його облікового запису). Оскільки ключ організації - це зашифровані доменом системи, її публічна IP адреса та порт, а домен буде частиною логіну користувача, то шляхом дешифрування ключа організації отримується загальна адреса API серверу ERP системи, із яким налаштовується на взаємодію API сервіс клієнтського додатку. На нього і надсилається запит на авторизацію. У разі успішної авторизації, користувач переходить до головної сторінки. А ні – користувач отримує повідомлення про проблему авторизації.

Після авторизації користувачеві надається можливість до всіх оперувати даними кожного доступного модуля. Для кожної таблиці кожного модуля постачальнику необхідно на програмному рівні реалізувати візуальні форми для заповнення та інші графічні елементи.

У даному додатку для кожної таблиці створюється три візуальні компоненти:

- DataGridView – відображення усіх даних таблиці у вигляді самої таблиці;

- CardView – картка конкретного запису таблиці із короткими відомостями про елемент та з переходом до форми елемента;

- FormView – форма елемента, де повною мірою розкривається інформація про нього та можливе доповнення додатковими даними. Також саме через форму відбувається створення нового елемента, його редагування або видалення.

Окрім цього, присутні дві окремі сторінки:

- DepartmentView – сторінка відділу працівника;

- ProfileView – профіль працівника, де він матиме можливість редагувати такі свої персональні дані як логін, пароль, номер телефону або адресу електронної пошти.

У момент закриття додатку система надсилає запит до серверу із операцією Disconnect, під час чого на сервері даний користувач видаляється із переліку онлайн користувачів.

8. Експеримент

Для визначення ефективності перетвореної програми проведені експерименти щодо кожного додатку системи.

Передовсім було проведено додаток головного веб-сайту. Процес реєстрації відбувається із використанням логіну, адреси електронної пошти та паролю. Одночасно відбувається валідація на наявність у системі користувача із вказаним логіном або імейлом. Процес авторизації відбувається із введенням логіну або імейлу та паролю. За умови успішної авторизації користувач потрапляє на сторінку свого облікового запису, де він може перейти до налаштувань свого профілю або до переліку своїх ERP систем.

На сторінці «Your ERP systems» користувач знайде усі свої створені ERP системи. У даній таблиці подається лише основна інформація (назва компанії, ключ організації, домен, тип бази даних). Розкриваючи кожний запис таблиці, можна додатково дізнатися адресу даної системи або перейти до веб-консолі вибраної ERP системи.

Наступним перевіряється додаток інсталятора. Налаштувати та запустити сервер системи можливо лише із використанням додатку інсталятора. Сам процес інсталяції складається із 4 кроків:

- Авторизація – користувач має бути зареєстрований у системі;
- Перевірка назви компанії – назва компанії має бути унікальна в межах сервісу;
- Налаштування серверу – необхідно вказати порт роботи серверу, а також вказати адресу підключення до бази даних;
- Вибір модулів, необхідних для встановлення.

Далі, переглянувши свої налаштування та підтвердивши їх, система реє-

струється на API та запускається на даному сервері. Після успішного запуску вказує користувачу відповідним повідомленням.

Потім відбувається перевірка роботоздатності сервера ERP системи. Перейшовши на порт веб-інтерфейсу, користувач переходить на його головну сторінку. Згідно з функціональними умовами, користувачу надається можливість оперувати відділами та обліковими записами працівників, а також переглядати записи усіх таблиць кожного модуля.

Останньою здійснюється перевірка роботоздатності клієнтського додатку для працівників. Запустивши клієнтський додаток, користувач має спочатку авторизуватись із використанням логіну, паролю та ключа організації. Успішно авторизувавшись, користувач потрапляє на головну сторінку, де можна побачити доступні йому модулі.

Перейшовши на сторінку відповідного модуля, користувач може відразу почати оперувати даними даного модуля.

Звісно, кількість даних при тестуванні була мала, але варто зазначити, що і реальні сервери будуть набагато потужніші за комп'ютер, на якому проводилися тестування (процесор Intel Core i7-8550K, 8 гігабайт оперативної пам'яті, 256 гігабайт SSD).

Висновки

Реалізовано сервіс систем управління ресурсів підприємства, що відповідає усім висунутим технічним та функціональним вимогам. На основі інструментарію та функціональних можливостей систем аналогів був сформований перелік технічних та функціональних вимог до проекту. На базі даних вимог були обрані технології та принципи проектування, які стануть у пригоді та полегшать процес розробки та тестування системи. За допомогою сучасних шаблонів проектування була розроблена архітектура системи, яка в перспективі може стати основою справжнього комерційного сервісу. Завдяки вищезазначеній концепції модульності дана система легко доповнюється та розширюється. В перспективі

сервіс можна доповнити впровадженням ERP систем не лише на сервері клієнта, а й у хмарі, як це пропонують компанії Oracle та Microsoft, а також доповнювати його новим функціоналом, залежно від вимог бізнесу.

1. Жураковський Ю.П. Теорія інформації та кодування: Підручник / Ю.П. Жураковський, В.П. Полторак. - К.: Вища шк, 2001. - 255 с.
2. Полторак, В.П. Інформаційна безпека та захист даних в комп'ютерних технологіях і мережах [Електронний ресурс]: навчальний посібник для студентів спеціальності 126 «Інформаційні системи та технології» / В. П. Полторак ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 1,77 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2020. – 78 с.
3. Top 3 Essential ERP Modules [Електронний ресурс]. – режим доступу: <https://www.chetu.com/blogs/technical-perspectives/oracle-erp-modules.php>
4. Что такое ERP и зачем это нужно? [Електронний ресурс] – Режим доступу до ресурсу: <https://dynamics.microsoft.com/ru-ru/erp/what-is-erp/>.
5. What is AES Encryption and How Does it Work? [Електронний ресурс]. – режим доступу: <https://searchsecurity.techtarget.com/definition/AdvancedEncryptionStandard>
6. Microsoft Dynamics 365 Business Central [Електронний ресурс]. – режим доступу: <https://dynamics.microsoft.com/en-cy/>
7. Oracle ERP [Електронний ресурс]. – режим доступу: <https://www.oracle.com/erp/>

References

1. Zhurakovskiy Y.P., Poltorak V.P. Theory of information and coding: Textbook, 2001. – p.10 (in Ukrainian).
2. Poltorak V.P. Information security and data protection in computer technology and networks [Online]: textbook for students majoring in 126 “Information Systems and Technologies”, 2020. – p.78 (in Ukrainian).
3. Top 3 Essential ERP Modules [Online] – Available from: <https://www.chetu.com/blogs/technical-perspectives/oracle-erp-modules.php>.

4. What is ERP and why it is necessary? [Online] – Available from: <https://dynamics.microsoft.com/ru-ru/erp/what-is-erp/> (in Ukrainian).
5. What is AES Encryption and How Does it Work? [Online]. – Available from: <https://searchsecurity.techtarget.com/definition/AdvancedEncryptionStandard> (in Ukrainian).
6. Microsoft Dynamics 365 Business Central [Online]. – Available from: <https://dynamics.microsoft.com/en-cy/>
7. Oracle ERP [Online]. – Available from: <https://www.oracle.com/erp/>

Отримано: 17.05.2022

Про авторів:

Тарасенко Артем Дмитрович, магістрант Національного Технічного Університету України «КПІ імені Ігоря Сікорського». <https://orcid.org/0000-0003-2631-2558>,
Дорошенко Анатолій Юхимович, доктор фізико-математичних наук, професор, завідувач відділу теорії комп'ютерних обчислень, професор кафедри інформаційних систем та технологій Національного Технічного Університету України «КПІ імені Ігоря Сікорського». Кількість наукових публікацій в українських виданнях – понад 190. Кількість наукових публікацій в зарубіжних виданнях – понад 80. Індекс Гірша – 6. <http://orcid.org/0000-0002-8435-1451>

Місце роботи авторів:

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», проспект Перемоги 37 та Інститут програмних систем НАН України, 03187, м. Київ-187, проспект Академіка Глушкова, 40.
Тел.: (044) 526 3559
E-mail: artemissterio@gmail.com, doroshenkoanatoliy2@gmail.com