https://doi.org/10.31891/2307-5732-2023-317-1-257-262
УДК 004.056.5:621

ПРАВОРСЬКА Наталія
Хмельницький національний університет
https://orcid.org/0000-0001-6001-3311
e-mail: margana2000007@gmail.com

ЗРУЧНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ БІЗНЕСУ З МОЖЛИВІСТЮ АДАПТАЦИИ ДО РИНКОВИХ ВИМОГ

Для підтримки повсякденної діяльності, бізнес-компанії потребують гнучких систем, які дозволять їм адаптуватися до ринків, що доволі швидко змінюються в умовах сьогодення. Однак, більша частина програмного забезпечення, яка існує на ринку, не задовольняє потреби конкретної компанії, не кажучи про адаптацію до вимог, які дуже швидко змінюються. Програмне забезпечення для підтримки підприємств, призначено в основному для вирішення поставлених конкретних завдань. Постає питання про можливість спрощення розробки програмного забезпечення для бізнесу, не маючи обмежень і також мати змогу створювати користувацьке програмне забезпечення саме для бізнесу не тільки для використання, а й для подальшої адаптації. Виникає необхідність уникнення багаторазового введення одних і тих самих даних у кілька звітних документів (створення звітів, оглядів, тощо) для зменшення накладних витрата запобіганню помилок.

Ключові слова: програмне забезпечення, бізнес-компанії, документація, Java Script, фреймворк, плагін, веб-сервер Apache, багатоцільове інтегроване середовище розробки (IDE) Eclipse, PHP-фреймворк Symfony.

PRAVORSKA Natalya
Khmelnitskyi national university

CONVENIENT SOFTWARE FOR BUSINESS WITH THE POSSIBILITY OF ADAPTATION TO MARKET REQUIREMENTS

To support day-to-day operations, business companies need flexible systems that will allow them to adapt to the markets that are changing quite quickly in today's environment. However, most of the software that exists in the market does not meet the needs of a particular company, let alone adapt to the requirements that change very quickly. Software for supporting enterprises, intended mainly for solving specific tasks. The question arises about the possibility of simplifying the development of business software, without limitations, and also being able to create custom software specifically for business, not only for use, but also for further adaptation.

In business-companies, there is a need to transmit external inspection data to laboratories, as well as the possible collection of these results and combining them into research reports, which contain the results of checks on the quality and quantity of manufactured products. Usually, in order to perform such actions, it is customary for companies to use standard office programs and paper records that will allow you to conduct reviews and create various reports. In such cases, it becomes necessary to enter the same data many times in several documents. Companies are trying to find applications that can reduce such overhead and prevent errors. Since there are no existing solutions to support such activities, there is a need to create such supporting software. The task is to develop a method called "administration generation" capable of introducing a level of abstraction to reduce repetitive tasks during development. Using the Apache open-source web-server, which allows customization for a specific business-company, and the Eclipse multipurpose integrated development environment (IDE), which supports multiple development languages, including PHP and JavaScript, this method was implemented. It used the fully pluggable Symfony PHP framework combined with the highly flexible Ext JS JavaScript framework. Symfony allows you to implement the right "administrative generation" property and extensibility to meet the right current market requirements. Ext JS is currently integrating to improve the user experience. Both the first and the second are known for their excellent documentation and active community.

Keywords: software, business-sompanies, documentation, Java Script, framework, Plugin, Apache web-server, Eclipse multipurpose sntegrated development environment (IDE), Symfony PHP framework.

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями

Сьогодні бізнес-компанії прагнуть не тільки використовувати гнучкі системи, які будуть забезпечувати їх повсякденну діяльність, але й мають допомагати швидко проводити адаптацію в таких умовах. Програмне забезпечення, яке існує на ринку може задовольнити виконання лише поставленої конкретної задачі, а не пристосовується під потреби конкретної компанії , не кажучи про спроможність швидкої адаптації до вимог. Є багато невеликих бізнес-компаній, які хотіли б отримати програмне забезпечення, яке можна не лише використовувати, а й підлаштовувати під свої проблеми. У бізнес-компаніях виникає необхідність передачі даних зовнішній інспекції, лабораторіям, а також можливій збір цих результатів та об'єднання їх в звіти про дослідження, які в собі містять результати перевірок на якість та кількість виробленої продукції. Зазвичай, для того, щоб виконати такі дії, у компаніях прийнято використовувати стандартні офісні програми та паперові записи, які дозволять проводити огляд та створювати різноманітні звіти. В таких випадках виникає необхідність введення багато разів одних і тих самих даних у кілька документів. Компанії намагаються знайти додатки, завдяки яким можливе зменшення подібних накладних витрат та запобігання помилкам. Оскільки існуючих рішень для підтримки такої діяльності не існує, виникає потреба у створенні подібного допоміжного програмного забезпечення.

Аналіз досліджень та публікацій

Багато бізнес-компаній на сьогодні намагаються провести адаптацію або просто обійти функціональність програмного забезпечення, замість того, щоб виконувати адаптацію додатку під потреби

компанії [1], те ж стосується і систем підтримки підприємств [2]. Через занадто високі витрати, компанії, які працюють з готовим комерційним програмним забезпеченням (наприклад, COTS – це упаковане або консервоване (готове) апаратне або програмне забезпечення, яке адаптоване на ринку до потреб закупівельної організації, а не введення в експлуатацію виготовлених на замовлення або індивідуальних рішень), майже ніколи не виконують адаптацію їх до своїх конкретних потреб [3]. Ті компанії, для потреб яких немає COTS-рішення, по всій ймовірності, застосовують для підтримки своєї діяльності лише стандартні офісні програми (наприклад, такі як електронні таблиці, текстові редактори і тому подібне). Однак через свою жорстку природу, додатки COTS, також ускладнюють для бізнес-компанії зміну її бізнес-процесів [3]. Створюється багато проблем тому, що програмне забезпечення не має змоги повністю підтримувати компанію, оскільки іде зростання накладних витрат, через надмірне повторення завдань та збільшення ймовірності появи помилок [4]. Через такі недоліки бізнес-компанії втрачають конкурентну перевагу та стають менш ефективними, ніж цього хотілося.

Виклад основного матеріалу

При вище описаних проблемах постає питання створення такого програмного забезпечення, корисних програм, які будуть підлаштовуватися під потреби діяльності і бізнес-процесів кожної окремої бізнес-фірми. Критерієм тут мають виступати підтримка робочого процесу компанії за відносно низьких витрат. Компанія, при підтримці подібного ПЗ не буде витрачати кошти та час на пошук обхідних шляхів, а зосереджується на своїй основній діяльності та підвищує свою ефективність. Для розробки індивідуального програмного забезпечення бізнесу , необхідно провести вивчення загальних бізнес-вимог, які стануть основою для пошуку методів та інструментів для полегшення реалізації виконання такого завдання.

В першу чергу компанії хочуть мати змогу проводити редагування представлення об’єктів (тобто клієнтів, продукції і тому подібне), з якими стикається бізнес-процес і з властивостями, з якими доводиться компанії працювати. Також виникає необхідність, і компанії зацікавлені в тому, щоб була можливість подання визначених даних у своїх звітах і оглядах, до того ж з можливістю їх адаптації під конкретно свій бізнес. Крім цього існує запит на обмін інформацією зі своєї системи з іншими компаніями та деякими стандартними додатками (наприклад, додатком для ведення бухгалтерського обліку, з яким компанії вже працюють, або додатком для роботи з електронними таблицями, для самостійної обробки своїх даних будь-яким зручним для них способом). Так як бізнес-компанії знаходяться в постійному розвитку, то допоміжне програмне забезпечення повинно мати змогу впоратися з такими змінами.

Ще існує спостереження, що, як керівний персонал, так і робітники бізнес-компанії хочуть мати доступ до своїх даних з різних місць і з різних пристроїв, а не тільки зі свого офісу, але й зі свого домашнього помешкання, чи коли вони знаходяться в готелі. Це буде створювати додаткові вимоги та потреби щодо безпеки, так як програмне забезпечення не зможе більше працювати автономно або в локальній мережі . і небажано, щоб дані цієї компанії біли у відкритому доступі.

Отже, провівши аналіз, з’ясовано, що для підтримки діяльності бізнес-компанії ПЗ має вміти:

- обробляти та маніпулювати різними даними бізнес-компанії, при цього підтримуючи їхній робочий процес;
- зберігати огляди цих даних (списки з декількох елементів; детально за елементами);
- підтримувати передачу цих даних (до різних користувачів; у різні компанії; до різних додатків);
- проводити адаптацію до змін всередині компанії.

Бажано, щоб програмне забезпечення мало наступні властивості:

- доступність (простота встановлення (для нових співробітників, на нові машини або у тимчасових місцях);
- легкість використання;
- доступність з будь-якого місця (дозволяє працювати з домашнього приміщення, готель, в дорозі);
- безпечність (недоступність для інших користувачів; не можливість відображення даних, для яких немає доступних облікових даних).

Для реалізації поставленої задачі розробки програмного забезпечення за вказаними параметрами було вирішено використовувати метод, який має назву «генерація адміністрування», спроможного вводити рівень абстракції для зменшення завдань, які повторюються під час розробки.

Перед початком розробки програмного забезпечення необхідно виконати аналіз роботи бізнес-компанії. Подібний аналіз можна проводити з використанням декількох методик:

- проведення співбесіди з робітниками та керівництвом компанії , для з’ясування їх проблем та побажань;
- потім проводиться аналіз документації компанії (такої, як звіти, огляди та тому подібне), щоб створити образ бізнес-процесу. Проведення додаткових інтерв’ю допомогло розумінню роботи компанії;
- інформація, яка отримана на основі інтерв’ю та документації об’єднувалась в один документ та розміщалася на сторінці компанії для перевірки та оновлення;
- проведення спостереження на протязі звичайного робочого дня в офісі бізнес-компанії необхідно для повного розуміння її робочого процесу. В ході спостережень було з’ясовано:

перелік та типи завдань, як часто завдання перериваються телефонними дзвінками від клієнтів та інших сторін для обміну інформацією.

В результаті проведення вище описаних кроків було отримано офіційний звіт про огляд, що містить дані з паперового носія (журналу) про кількість (можливо якість і інше) продукції, які клієнти використовують для остаточної оплати. Також виникає необхідність з'ясовування акторів (тобто контакти самої бізнес-компанії з різними сторонами). Як виявилося з цього акторами виступає сама компанія (а також її головний офіс); клієнти (ті, хто перевіряє якість отриманої продукції та проводить оплату); сюрвеєри (страховики, які виконують огляд та надають оцінку якості транспорту та товару, який перевозиться); лабораторії (які аналізують зразки, щоб перевірити якість продукції); агенти (які призначаються відправниками і забезпечують спосіб зв'язку між відправником і людьми на землі); термінали (місця для перевантаження продукції); страхувальники (які вирішують проблемні ситуації та це організовано у співпраці із клієнтом та сюрвеєром); бухгалтер (допомагає вести бухгалтерію бізнес-компанії.

Наступним кроком буде складання всіх документів, які необхідні компанії та створення зведеної бази даних, де можна буде об'єднати дані та звіти, куди вони будуть заноситися. Для встановлення вимог для допоміжного програмного забезпечення постає також необхідність в проведенні аналізу всього робочого процесу бізнес-компанії. Після цього створюються таблиці з усіма завданнями робочого процесу та агентами, Таким чином проводиться зв'язок між тим, хто за яке завдання буде відповідати. Можна розробити діаграми для наочного показу завдань та агентів. Після опитування співробітників виконується корекція та оновлення представлених таблиць та діаграм моделей.

Наприклад, можна розглянути інформацію про створення звітів для роботи бухгалтерії, яка представлена в таблиці 1. Відповідно цієї таблиці можна наочно побачити послідовність виконання завдання на діаграмі рисунок 1. Також це стосується і завдання для самої бізнес-компанії, яке представлене в таблиці 2 та його наочне представлення на рис. 1 і рис. 2, відповідно.

Таблиця 1.

Послідовність виконання завдань робочого процесу пересилки продукції компанії

Активність	Створити остаточний звіт
Завдання	1. Отримати присвоєння бізнес-компанії 2. Перевірити всю інформацію 3. Об'єднати інформацію у звіті 4. Надіслати клієнту та бізнес-компанії 5. Створити рахунок 6. Позначити завдання як виконане, щоб видалити його з потоку завдань

Таблиця 2.

Послідовність виконання завдань робочого процесу компанії відносно клієнта

Активність	Створити рахунок
Завдання	1. Отримати присвоєння бізнес-компанії 2. Перейти до рахунку 3. Перевірити розрахунки 4. Надіслати клієнту та бізнес-компанії

Після того, як будуть описані всі етапи робочого процесу бізнес-компанії та її діяльність ретельно проаналізована, можна приступити до визначення та опису проблеми, яка постає перед компанією. Незважаючи на те, що діяльність бізнес-компанії може мало відрізнятися від подібних їй, можуть існувати унікальні моменти.

Саме з цієї причини, якщо навіть і є програмне забезпечення, яке допомагає у робочому процесі бізнес-компанії для виготовлення своєї продукції, його результати необхідно прив'язувати до документації при подальшій роботі відділів компанії (наприклад, бухгалтерії, відділу постачання і тому подібне)

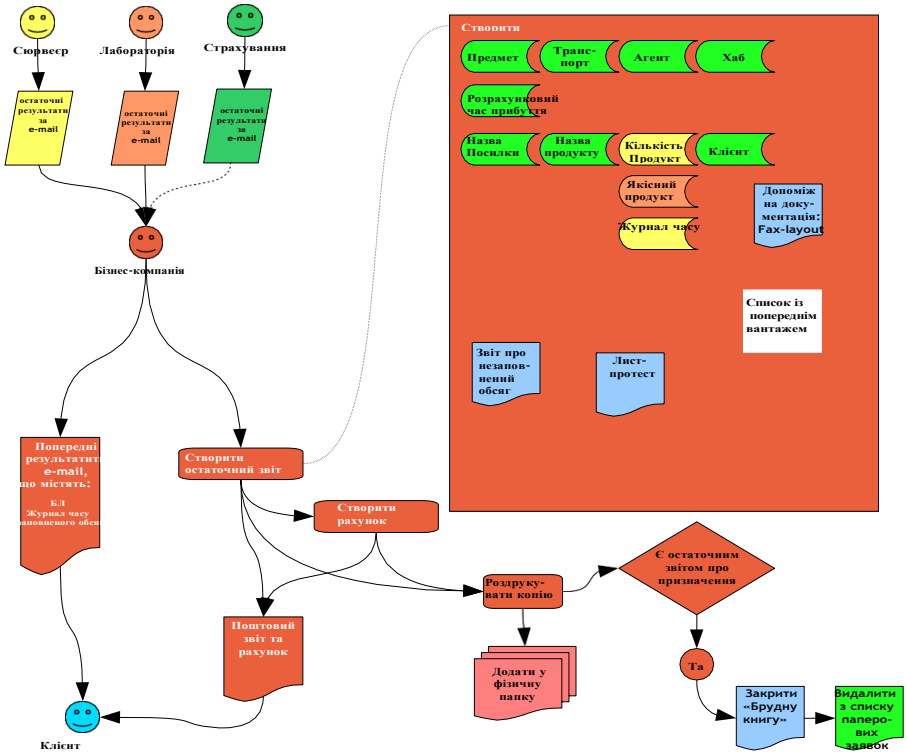

Рис. 1. Діаграма завдань робочого процесу, після опитування робітників бізнес-компанії

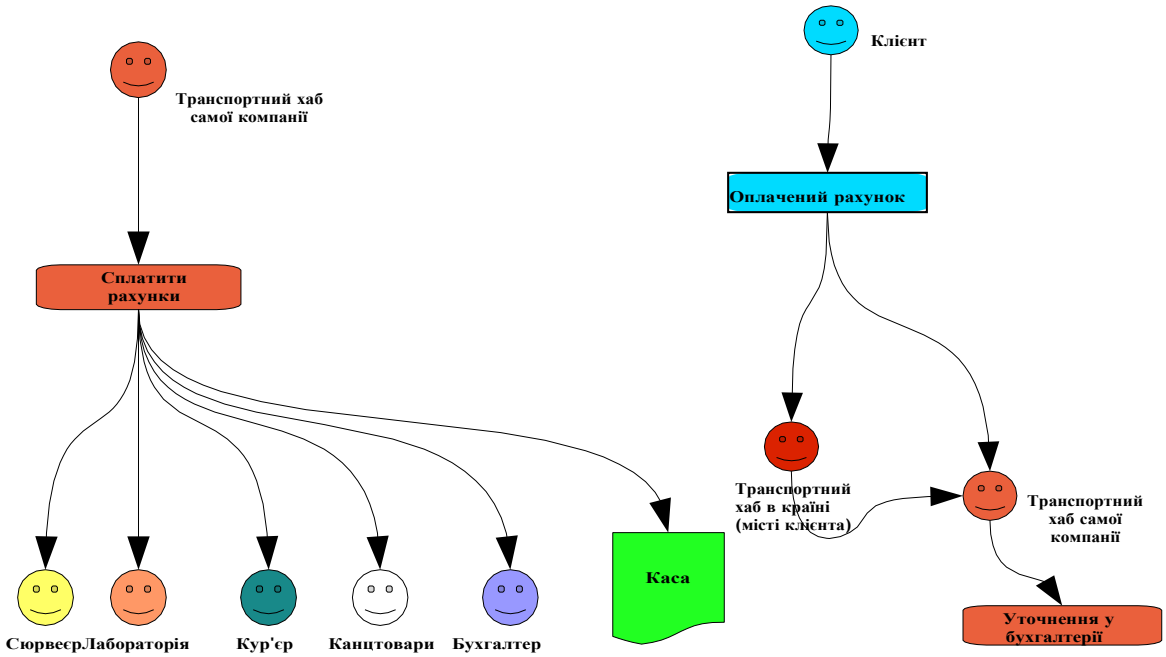

Рис. 2. Завдання робочого процесу компанії, пов'язані з грошовим потоком

Отже зазвичай готових COTS-рішень не існує і компанія має витрачати кошти на замовлення програмного забезпечення, яке буде задовольняти всі її потреби, що робить його непривабливим для софт-верних компаній, які будуть займатися вирішенням такої проблеми.

На даний момент багато невеликих бізнес-компаній, в якості підтримки свого бізнесу задовольняються стандартними офісними програмами, такими як Microsoft Office (Microsoft Word, Excel і Access) та поштових клієнтів (наприклад, Mozilla Thunderbird, Gmail, Microsoft Outlook, тощо). В такому разі накладні витрати збільшуються, оскільки вони проводиться виконання великої кількості завдань для оновлення всіх своїх оглядів, які можуть повторюються в документах і навіть в одному документі , але в різних місцях. Після виконання всієї роботи, оновлення знову розсилають різним сторонам, при чому вибір документів іде в ручному режимі та розсилання проводиться відповідним компаніям та клієнтам. Тому помилки тут реально становлять високу ймовірність і як, правило, іде зростання накладних витрат.

Однак, на цьому проблеми не закінчуються. Комп'ютери в бізнес-компаніях залежать також своєї потужності та своєї захищеності від вірусів. Тому в деяких випадках компанії проводять дублювання своїх

даних в паперових варіантах (журналах, звітах і тому подібне). В цьому випадку накладні витрати будуть зменшуватися при довірі до комп'ютерів.

При роботі над розв'язанням проблеми було вирішено скористатися методом «генерації адміністрування», який дає змогу вводити рівень абстракції для зниження кількості завдань, які будуть повторюватися під час розробки програмного забезпечення. Однак цей метод не може творити «чудеса». Можливості «генерації адміністрування» обмежуються зумовленою функціональністю, яка буде визначатися у шаблонах, при поєднанні з інформацією, отриманою з моделі даних та конфігурації. В такому випадку, це буде означати існування дозволу маніпулювання конкретними елементами відповідно до моделі даних та відображення огляду цих елементів.

Якщо можливості генератора занадто обмежені, то після того, як буде згенеровано код програмного продукту, його можна не тільки розширити, а й покращити і навіть замінити на той код користувача, який буде дозволяти представити результат в такому вигляді, який необхідний для виконання завдань робочого процесу. Однак, незважаючи на те, що такий підхід робить створення адміністрування доводі гнучким інструментом, треба пам'ятати, що заміна згенерованого коду або навіть встановлення великої кількості властивостей у конфігурації треба використовувати з великою обережністю, оскільки це може сварити проблему подібну до створення шаблонів. Це буде ускладнювати підтримку відповідності застосунку до моделі даних у разі її зміни. Дуже добре, що змін, які потребуються лише для того, щоб дії та представлення відповідали новій моделі даних набагато менше, тому їх реалізація доволі швидка та проста.

Генерація адміністрування використовується для:

1. Створення на основі моделі даних та конфігурації представлень для об'єкта (можливість користувацького коду розширювати, покращувати та змінювати згенерований код; складність програмування користувацького коду, як і «звичайного» коду);

2. Використання паттернів для шаблонів генератора (скорочення часу розробки та зміни при помилках; відносно легка адаптація до змін у моделі даних).

Недоліком генерації адміністрування можна назвати необхідність часу, для звикання написання шаблонів.

Метод «генерації адміністрування» було реалізовано при використанні веб-сервера з відкритим кодом Apache, який дає змогу налаштування під потреби конкретної бізнес-компанії та багатоцільового інтегрованого середовища розробки (IDE) Eclipse, що підтримує кілька мов розробки, включаючи PHP і JavaScript, даний метод було реалізовано.

Використовувався PHP-фреймворк Symfony, що повністю підключається, у поєднанні з дуже гнучким JavaScript-фреймворком Ext JS. Symfony дозволяє реалізувати потрібну властивість "генерації адміністрування" та розширення для відповідності потрібним сьогоденним ринковим вимогам. Ext JS в цей час виконує інтеграцію для поліпшення взаємодії з користувачем. Як перший, так і другий відомі своєю відмінною документацією і активною спільністю.

Symfony – це безкоштовна серверна PHP-інфраструктура з відкритим вихідним кодом, яка повністю підключається і підтримує об'єктно-орієнтовану гнучку (веб-розробку). Це зводиться до того, що вона використовує найкращі практики, доступні для методів, таких як шаблон ModelViewControl (MVC), щоб розділити знання та логіку їхнього подання. Надається повністю об'єктно-орієнтоване середовище, в якому об'єктно-реляційне відображення (ORM) використовується для прозорого перетворення даних між (PHP) об'єктами та таблицями в реляційній базі даних. Нарешті, доступні інструменти формування шаблонів та адміністрування для автоматичного створення дій та уявлень на основі моделі. Генератор адміністрування, як і решта фреймворку, можна настроювати, змінювати та розширювати.(за допомогою плагінів), щоб змусити його робити все, що завгодно. Все це допомагає скоротити час розробки та дозволяє створювати продукт, орієнтований на користувача.

Однією з найважливіших функцій, інтегрованих у Symfony, безперечно є ORM. У Symfony зараз доступні дві різні ORM: Propel і Doctrine. Обидва ORM доступні для PHP, і там, де Propel в даний час інтегрований за умовчанням у Symfony, Doctrine можна встановити як плагін, щоб повністю замінити Propel. У Propel є кілька завдань у Symfony. Як і слід очікувати від ORM, він пропонує можливість прозорого доступу та управління даними, що зберігаються в базі даних, з об'єктів у середовищі розробки. І тому він генерує базові класи з урахуванням таблиць, визначених у базі даних. Створюються так звані однорангові класи, які містять інформацію про те, як витягувати елементи з бази даних та поміщати їх у сконструйовані об'єкти. Нарешті, надаються класи-оболонки, які розширюють ці базові об'єкти та базові однорангові класи, щоб запропонувати можливість покращити їх за допомогою функцій користувача.

Крім створення цих класів, Propel також може створювати структуру самої бази на основі опису моделі даних. За замовчуванням цей опис має бути написаний на XML, але Symfony надала додатковий рідер, здатний конвертувати YAML в XML, що робить визначення цього опису дуже простим (рис. 3). Інший спосіб створення класів можливий шляхом надання Propel вже існуючої бази даних, щоб він міг базувати визначення моделі безпосередньо з бази даних.

```yaml
propel:

  # countries
  country:
    country_id:
      type: INTEGER
      required: true
      autoIncrement: true
      primaryKey: true
    name:
      type: VARCHAR
      size: 150
      required: true
    abbreviation:
      type: VARCHAR(4)
      required: false


  # cities
  city:
    city_id:
      type: INTEGER
      required: true
      autoIncrement: true
      primaryKey: true
    country_id:
      type: INTEGER
      required: true
      index: true
      foreignTable: country
      foreignReference: country_id
      onDelete: RESTRICT    # Ви не можете видалити країну, поки є пов'язане з нею місто
      onUpdate: CASCADE
    name:
      type: VARCHAR(150)
      required: true
```

Рис. 3 Приклад визначення моделі даних у файлі schema.yml

Саме властивості ORM у Symfony, такі як: можливість налаштування бази даних, створення таблиць зі стовпцями на основі опису моделі даних; можливість створення об'єктної моделі з одноранговими класами для отримання декількох елементів на основі моделі даних або існуючій базі даних; дозволяє налаштовувати об'єкти з об'єктної моделі, тому можна реалізувати всі бізнес-процеси і логіку; дозволяє налаштовувати однорангові об'єкти, що супроводжують об'єктну модель, щоб мати можливість визначати розширені відносини та для налаштування продуктивності; ORM Propel, що підключається, за замовчуванням можна замінити іншими, такими як Doctrine – дають змогу в процесі конструювання програмного забезпечення вирішувати поставлені завдання перед розробниками.

Висновки з даного дослідження і перспективи подальших розвідок у даному напрямі

На сьогодні постає питання в допомозі невеликим бізнес-компаніям для спрощення розвитку бізнесу, отримати програмне забезпечення, яке при цьому буде давати змогу, не тільки використання в робочому процесі, в й адаптації до нових ринкових вимог. Тобто, компанія повинна мати можливість ефективного використання створеного програмного застосунку за своїми потребами (це стосується вимог ринкових, платіжних, митних, податкових і таке інше). Є намагання за допомогою подібних програмних застосунків зменшити накладні витрати та уникнути помилок, через багаторазове введення одних і тих самих даних в різні документи, які компанія має надавати не тільки клієнту, а в різні установи.

На сьогодні майже не має готових рішень для підтримки таких вимог, тому після проведення аналізу робочого процесу окремої бізнес-компанії є спроба розробити такий програмний продукт з використанням, як методу «генерації адміністрування», так і інтегрованого середовища та PHP-фреймворку. Представлена тільки невеличка модель робочого процесу, за якою будуть створюватися діаграми моделі даних, а потім на її основі буде написаний код програмного забезпечення. Також використання такого потужного інструменту, як Symfony, дасть змогу сконструювати програмне забезпечення, яке б задовольняло всі поставлені замовником вимоги.

References

1. Wallace Thomas F., Kremzar Michael H. ERP: Making It Happen (3rd Ed.) Wiley, Hardcover – 2001– 384 p.
2. Ree, 2007: Leon van der Ree, Enterprise Support Systems, 2007
3. Themistocleous, Irani, O'Keefe and Paul, Marinos Themistocleous, Zahir Irani, Robert M. O'Keefe and Ray Paul.: ERP Problems and Application Integration Issues: An Empirical Survey – Proceedings of the 34th Annual Hawaii International Conference on System Sciences – 2001.
4. Bouwman, Hooff, Wijngaert & Dijk, 2005: Harry Bouwman, Bart vd Hooff, Lidwien vd Wijngaert and Jan A G M v Dijk, Information & Communication Technology in Organizations, 2005, ISBN: 1412900905
5. Ronald E. Rise and Paul M. Leonardi. Information & Communication Technology in Organizations. – The SAGE Handbook of Organizational Communication: Advances in Theory, Research, and Method – SAGE Publications, 2013 – 848 p.