

УДК 004.02: 004.04.321

МОДЕЛЮВАННЯ АРХІТЕКТУРНИХ РІШЕНЬ ПІДТРИМКИ МУЛЬТИСАЙТОВОСТІ ДЛЯ ОРГАНІЗАЦІЇ ІНФОРМАЦІЙНИХ СИСТЕМ

А. В. Заїка, М. І. Філенко, А. С. Остапченко, Т. А. Григорова

Кременчуцький національний університет імені Михайла Остроградського
вул. Першотравнева, 20, м. Кременчук, 39600, Україна. E-mail: anton_zaiika@rambler.ru

Запропоновано архітектуру для підтримки мультисайтовості соціальної платформи – інформаційної системи, яка б дозволяла організувати зворотній зв'язок громадськості з виконавчою владою за рахунок надання можливості кожному активному члену спільноти, які не байдужі до соціальних проблем, подавати, обговорювати, підтримувати пропозиції щодо їх вирішення. У процесі роботи над соціальною платформою – інформаційною системою – було проаналізовано існуючі технології та архітектурні стилі для проектування складних інформаційних систем, розглянуто різні моделі і методи реалізації інформаційних систем. Для реалізації поставленої задачі було використано підхід, який є комбінацією відомих підходів, що використовуються при проектуванні та розробці інформаційних систем такого рівня. Спроектовано модель даних і діаграму класів, які повністю відповідають вимогам web-ресурсу.

Ключові слова: архітектурний стиль, мультисайтовість, інформаційна система, моделювання.

МОДЕЛИРОВАНИЕ АРХИТЕКТУРНЫХ РЕШЕНИЙ ПОДДЕРЖКИ МУЛЬТИСАЙТОВОСТИ ДЛЯ ОРГАНИЗАЦИИ ИНФОРМАЦИОННЫХ СИСТЕМ

А. В. Заика, М. И. Филенко, А. С. Остапченко, Т. А. Григорова

Кременчугский национальный университет имени Михаила Остроградского
ул. Первомайская, 20, г. Кременчуг, 39600, Украина. E-mail: anton_zaiika@rambler.ru

Предложена архитектура поддержки мультисайтовости социальной платформы – информационной системы, позволяющей организовать обратную связь общественности с исполнительной властью за счет предоставления возможности каждому активному члену общества, не равнодушному к социальным проблемам, подавать, обсуждать, поддерживать предложения по их решению. В процессе работы над социальной платформой – информационной системой – были проанализированы существующие технологии и архитектурные стили для проектирования сложных информационных систем, рассмотрены различные модели и методы реализации информационных систем. Для реализации поставленной задачи был использован подход, представляющий собой комбинацию известных подходов, используемых при проектировании и разработке информационных систем такого уровня. Спроектированы модель данных и диаграмма классов, полностью отвечающих требованиям web-ресурса.

Ключевые слова: архитектурный стиль, мультисайтовость, информационная система, моделирование.

АКТУАЛЬНІСТЬ РОБОТИ. На сьогоднішній день у зарубіжних країнах світу, таких як Франція, Італія, Німеччина, накопичений значний досвід дії різних моделей місцевого самоврядування, за допомогою яких відбулася трансформація місцевої влади, що зумовило покращення життя населення цих країн. Щодо України, то можна сказати, що сучасний процес демократичних перетворень в українському суспільстві вимагає від держави вирішення нових правових, економічних, політичних і соціальних завдань, головними з яких є формування ефективної системи місцевого самоврядування як максимально наближеного до населення інституту, задоволення інтересів громадян в усіх сферах життєдіяльності, впровадження дієвих механізмів соціальної взаємодії з самоврядними структурами. На даний момент громадянин не має ефективного, швидкого і дієвого інструменту для впливу на управління суспільством. Відсутня послідовна прозора система реалізації вимог та ініціатив громадян, яка базувалася б на актуальній об'єктивній думці суспільства. Відслідковування розгляду і виконання поданих пропозицій органами влади ускладнюється в силу того, що потребує багато часу та зусиль. У комерційних і приватних організаціях профспілки частіше за все відсутні, а якщо й є, то віддалені від працівників, вони не захищають їх інтереси і не відображають їх думку з приводу покращення умов праці. Органи влади, в свою чергу, планують свою

роботу не завжди враховуючи потреби громадян.

Сучасні телекомунікаційні технології дають змогу громадянам впливати на управління державними структурами дистанційно, витрачаючи мінімум часу і зусиль. Наприклад, такі закордонні системи, як Common town та American petitions дозволяють за допомогою форумів та приватного листування організувати взаємодію між людьми (працівниками та керівництвом), що дає змогу структурувати та налагоджувати роботу закладів. Також можна розглянути соціальні мережі такі, як Facebook та інші, які дозволяють спілкуватися і створювати спільноти за інтересами; серед вітчизняних систем такі, як сайт запитів «Доступ до правди» та громадянська платформа «Нова країна». Перший спрощує отримання доступу до необхідної інформації за рахунок організації запитів, другий дає можливість подання ініціатив. Але, що стосується втілення у життя онлайн моделі самоврядування, не один із цих сайтів цю задачу не вирішує. Повноцінний ресурс повинен давати змогу не тільки формувати свої пропозиції та підтримувати інші, але й формувати навколо них ініціативні групи, включаючи громадські організації, які б просували ці пропозиції, контролювали їх виконання та інформували про це громаду. Користувачі такої системи безпосередньо зможуть брати участь у процесі розвитку населеного пункту місця проживання або підприємства, компанії, де вони працюють. Це дозволить зменшити бюрократичну

та корупційну складову виконавчої влади, керівництва підприємства, що є дуже актуальним, як для України, так і для багатьох країн світу, а також буде сприяти розвитку суспільства. Виходячи з цього, можна зробити висновок що суспільству бракує ефективного, повнофункціонального онлайн-інструментарію для ефективної роботи з виконавчою владою та підтримки роботи ініціативних груп і громадських організацій.

Щоб забезпечити надійність функціонування такої складної інформаційної системи виникає необхідність у ретельному виборі або моделюванні її архітектури та проектуванні. Можна сказати, що моделювання архітектури проекту – це процес формування структурованого рішення, яке відповідає всім технічним й операційним вимогам. Невірне визначення ключових сценаріїв або невірне проектування загальних питань, нездатність виявити довгострокові наслідки основних рішень можуть поставити під загрозу функціонування всієї системи [1–3]. Саме адекватна архітектурна модель та її послідовна реалізація в процесі розробки інформаційної системи забезпечує логічну цілісність проекту, що забезпечує його успіх [2–4]. Підвищення вимог до обґрунтованості та якості архітектурних рішень є причиною зростання уваги дослідників у галузі інформатики до проблем архітектурного проектування інформаційних систем [5].

Метою роботи є моделювання архітектурного рішення для створення соціальної платформи інформаційної системи, яка б дозволяла організувати зворотній зв'язок громадськості з виконавчою владою за рахунок надання можливості кожному активному члену спільноти, які не байдужі до соціальних проблем, подавати, обговорювати, затверджувати пропозиції щодо їх вирішення.

МАТЕРІАЛ І РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ. Головною задачею при побудові платформи є забезпечення можливості створення необмеженої кількості підсистем (майданчиків), які б мали гнучку структуру адміністрування для налаштування їх на будь-якій тип організації, спільноти від підприємств, територіальних громад до країни в цілому.

На сьогоднішній день існує велика кількість архітектурних стилів, а саме: клієнт/серверний (система поділяється на два додатки, де клієнт виконує запити до сервера), компонентний архітектурний стиль (функціональні або логічні компоненти з можливістю повторного використання), N-рівнева/трирівнева архітектура (функціональність виділяється в окремі сегменти), об'єктно-зорієнтована та сервісно-зорієнтована архітектури. Архітектура програмного забезпечення практично ніколи не обмежена лише одним архітектурним стилем, часто вона є поєднанням архітектурних стилів, що утворюють повну систему. Поєднання архітектурних стилів корисно при побудові Інтернет веб-додатків, де можна досягти ефективного поділу функціональності за рахунок застосування багатопланового архітектурного стилю.

Якщо розглянути Facebook, який є мабуть найбільш обговорюваним інтернет-проектom у всьому світі, то стає зрозуміло, щоб справлятися з таким величезним соціальним графом і задовольняти пот-

реби у спілкуванні неймовірно великої кількості людей по всьому світу необхідно мати міцну архітектурну модель. Провівши аналіз були визначені її основні принципи. Архітектура Facebook – це трирівнева архітектура, що складається із таких слоїв, як веб-додаток, розподілений індекс та постійне сховище інформації. Для хешування даних, генерація яких вимагає великої кількості ресурсів використовується Memcache, для зберігання необхідної інформації – MySQL, для взаємодії між сервісами – Thrift – інтерфейс, для збору та агрегації даних із робочих серверів – універсальна система Scribe. Для кожного сервісу системи існує своя архітектурна модель, наприклад архітектурна модель пошуку інформації, що приведена на рис 1.

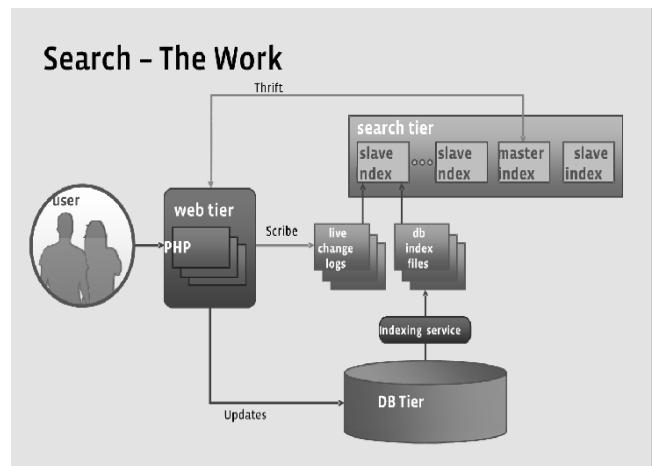


Рисунок 1 – Архітектурна модель пошуку інформації у Facebook

Web-ресурс, що розробляється відрізняється від соціальних мереж таких, як Facebook тим, що він має чітке спрямування, яке направлено на розвиток самоврядування, але водночас підтримує такі функції, як можливість створення особистих кабінетів, спілкування і обговорення. Крім того, для розвитку самоврядування система буде мати можливість розширення за рахунок організації підсистем (майданчиків), що будуть створюватися для різних типів організацій і будуватися на базі однієї глобальної платформи. Для реалізації цієї ідеї було запропоновано підхід на основі мультисайтовості, який полягає у використанні єдиної бази даних, єдиних шаблонів із можливістю налаштування кожної окремої підсистеми зі своїм доменним ім'ям, що будується на основі єдиної платформи. Схожа технологія використовується в системах управління контентом (CMS – Content Management System), які вирішують завдання адміністрування, управління та підтримки функціональності сайту. Але основною задачею CMS є використання готових шаблонів для побудови сайтів та управління кожним із них як окремим веб-додатком, які не пов'язані між собою [6–9].

Для реалізації поставленої задачі було спроектовано трирівневу клієнт-серверну архітектуру на базі об'єктно-зорієнтованого підходу. В архітектурі клієнт-сервер (рис. 2) програмний додаток моделюється як набір сервісів, що надаються серверами, й

безліч клієнтів, що використовують ці сервіси. Клієнти знають про доступні (наявні) сервери, хоча можуть і не мати уявлення про існування інших клієнтів. У нашому випадку сервісом виступає головна мультисайтова платформа, яка надає послуги створення необмеженої кількості підсистем, які мають гнучку структуру налаштування. Клієнтами, в свою чергу, є будь-який тип організації, спільноти, вид підприємства, територіальних громад до країни у цілому.

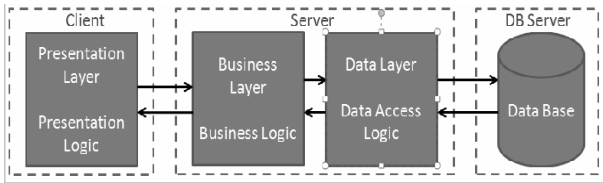


Рисунок 2 – Трирівнева клієнт-серверна архітектура

Але у той же час кожна підсистема є водночас незалежною ланкою в системі і має свій сервіс і своїх клієнтів, тобто архітектура проекту ускладнюється за рахунок мультисайтовості. Отже кожна підсистема також має клієнт-серверну архітектуру. В даному випадку сервісом будуть виступати послуги, які надають підсистеми інформаційної системи, а саме: будь-яка особа, зареєстрована в підсистемі

може пропонувати ініціативу, висловлювати скаргу або декларувати опитування спільноти (пропозиції); створювати ініціативні групи навколо запропонованої ініціативи, які будуть брати на себе відповідальність за втілення її в життя, організувати зв'язок між спільнотою та виконавчою владою та контролювати процес виконання ініціативи; за рахунок модерації виконувати перевірку поданих ініціатив, скарг й опитувань з точки зору нелітературної лексики та коректності висловлювань; підтримувати публікацію новин тощо.

Отже, у спроектованій моделі клієнт запитує сервіси тільки в сервера, але не в інших клієнтів; сервери можуть функціонувати як клієнти й запитувати сервіси в інших серверів, але не в клієнтів; клієнти повинні знати про сервіси, надавані певними серверами, і про те, як взаємодіють ці сервери. Головними перевагами є те, що архітектура досить відкрита і дозволяє за необхідності додавати в систему нові ресурси. Щоб забезпечити гнучкість і масштабованість системи та щоб упоратися із системним навантаженнями, можна створювати екземпляри системи з однаковими сервісами, які будуть надаватися різними об'єктами або різними екземплярами об'єктів. Організація вкладеної архітектури виглядає наступним чином (рис. 3).

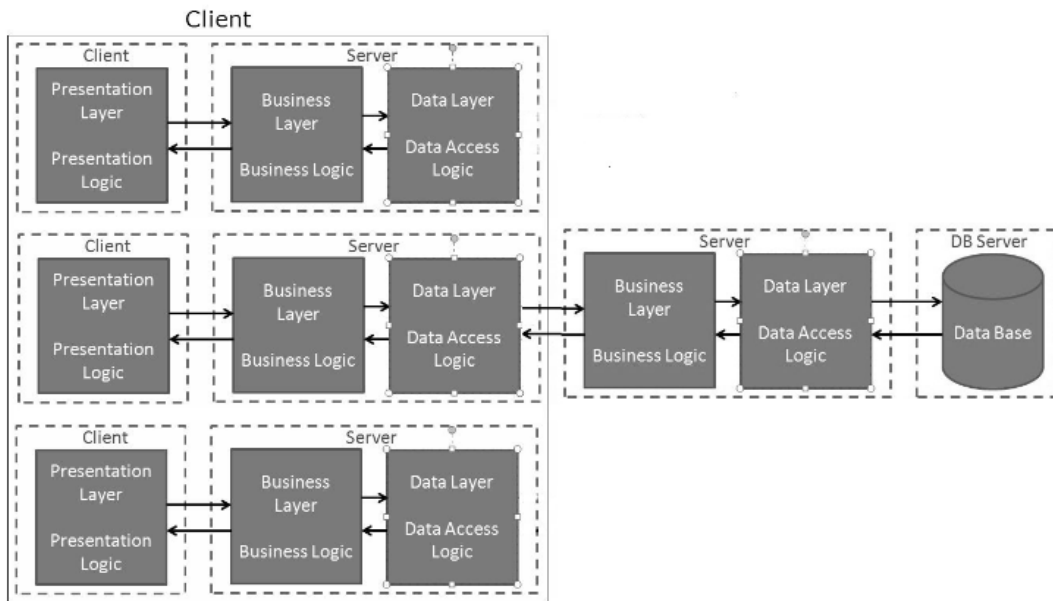


Рисунок 3 – Вкладена клієнт-серверна архітектура

Для того, щоб відділити бізнес-логіку проекту від її візуалізації, було використано шаблон проектування MVC (Model-View-Controller).

Модель даних є основою для інформаційної системи. При розробці моделі даних проекту було використано підхід побудови мультисайтовості з використанням однієї бази даних, що дозволило уникнути клонування БД при створенні нової підсистеми і, таким чином, зменшити кількість активних підключень до СУБД одночасно, а також забезпечило гнучкість проекту стосовно оновлення та внесення змін до БД.

Щоб спростити розробку моделі даних її умовно поділили на дві частини глобальну і модель даних для підсистеми.

Структура першої складається із таких таблиць: `global_platforms` – список усіх майданчиків, статус, ім'я, домен і користувача; `global_transactions` – статус транзакції, ім'я створювача транзакції, час оплати; `global_posts` – новини головного сайту, час публікації, створювача публікації та зміст новини; `global_users` – інформація про користувачів, а саме: ім'я, пароль, логін, фамілія, пошта, город, регіон, хобі. Зв'язки між таблицями зображено на рис. 4.

Другий модуль – Platform відповідає за роботу кожної підсистеми, що є складовою частиною глобальної платформи та їх сервіс (рис. 7).

Складається із таких класів: DefaultContrtoller – контролер, який відповідає за головну сторінку платформи. Показ новин на цій сторінці та іншу інформацію. GroupController – контролер, який відповідає за ініціативні групи (створення, обмін повідомленнями між усіма учасниками групи, інформація про групу та інше). PollsContrtoller – контролер, який відповідає за опитування (створення та налаштування варіантів опитування, підрахунок голосів).

ComplaintsContrtoller – контролер, який відповідає за скарги (створення, вивід на екран). InitiativesContrtoller – контролер, який відповідає за ініціативи та їх контроль (контроль завдань, які потрібно виконати для досягнення мети, статус ініціативи). AdminContrtoller – відповідає за модерацію платформи. Модераторів підсистеми призначає її власник. Модератори мають можливість додавати новини, робити будь-які налаштування платформи (бланки для ініціатив, список статусів для ініціатив, публікація новин, видалення неправомірного контенту з платформи тощо).

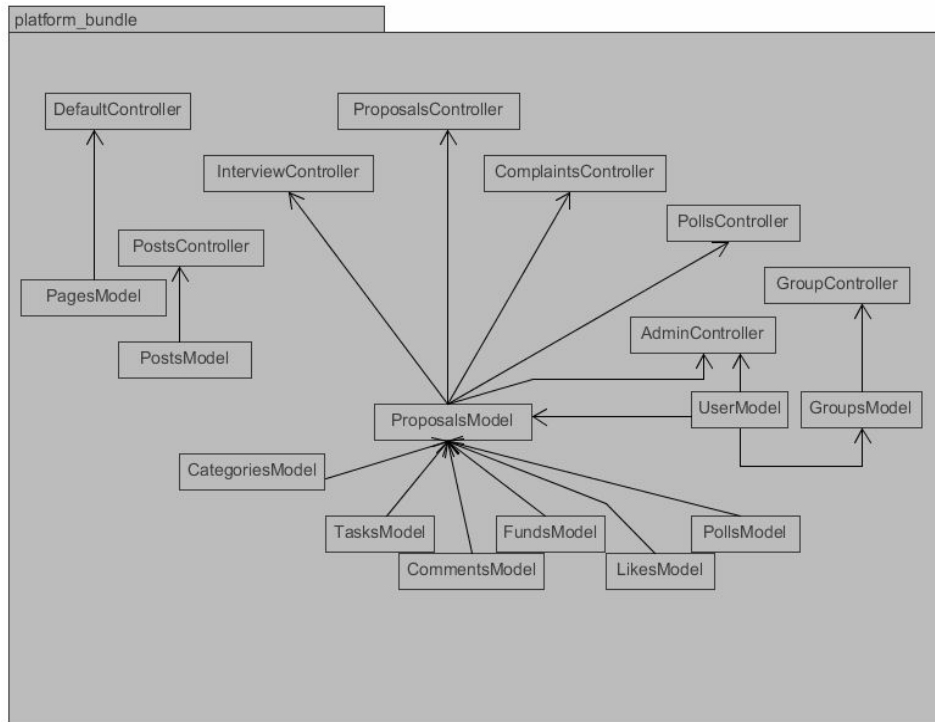


Рисунок 7 – Діаграма класів PlatformBuddle

ВИСНОВКИ. В ході роботи було ретельно проаналізовано предметну область, підходи щодо реалізації складних інформаційних систем, що працюють в інтернет просторі й їх архітектурні рішення.

Запропоновано архітектурне рішення, яке забезпечує підтримку мультисайтовості для соціального проекту інформаційної системи, що підтримує організацію зворотного зв'язку громадськості з виконавчою владою. Розроблено проект цієї інформаційної системи.

ЛІТЕРАТУРА

1. Фаулер М., Райс Д., Фомелл М., Хайт Э., Ми Р., Стаффорд Р. Архитектура корпоративных программных приложений / Пер. с англ. – М.: И.Д. «Вильямс», 2006. – 544 с.
2. Основи розробки веб-додатків: навчальний посібник // В.В. Осадчий, В.С. Круглик. – Мелітополь: ТОВ «Видавничий будинок ММД», 2012. – 540 с.
3. Батюк А.Є., Михайлюк О.Л. Система Автоматизованої розробки програмного забезпечення з трьохрівневою архітектурою // Праці міжна-

родних конференцій комп'ютерних наук та інформаційних технологій, 27–29 вересня 2007 р., м. Львів. – С. 179–182.

4. Розподілені системи. Принципи та парадигми // Е. Таненбаум, М. ван Стеен. – СПб.: Пітер, 2003. – 877 с.

5. Джеймі Сірович, Крістіан Даруй. Пошукова оптимізація сайту (SEO) на PHP для професіоналів. // Керівництво розробника по SEO. Професійний Пошук. – М.: Діалектика, 2008. – 352 с.

6. Томсон Л., Веллінг Л. Розробка веб-додатків на PHP і MySQL. – М.: Вільямс, 2009. – 330 с.

7. Артур Рим. Сучасні системи управління контентом. – СПб.: Питер, 2009. – 152 с.

8. Картузов А.В. Использование образовательных ресурсов Интернет // Кооперативная самобытность в новом тысячелетии: Тез. докл. межвуз. науч.-практ. конф. – Чебоксары, 2001. – 215 с.

9. Джон Вандюк. CMS Drupal 6: система для создания сайтов и управления ими – 2-е издание. – М.: «Вильямс», 2009. – 576 с.

DESIGN OF ARCHITECTURE FOR SUPPORT MULTI-SITE IN INFORMATION SYSTEMS

A. Zaika, M. Filenko, A. Ostapchenko, T. Hryhorova

Kremenchuk Mykhailo Ostrohradskyi National University

vul. Pershotravneva, 20, Kremenchuk, 39600, Ukraine E-mail: anton_zaika@rambler.ru

The article describes the architecture for support multi-site in information systems. The concept of the proposed solution is to create a fully functional universal system of online management community, which is defined by these participants. This platform let users to propose social initiatives for public consideration, to organize initiative groups for realization of these ideas, to get fresh information about initiatives' status. Technologies that exist and the architectural styles were considered in the design process of social information system. Different models and methods for implementation of information systems were also considered. For the specified task, it was used a method, which is a combination of the known methods of designing and developing information systems of such level. Also, it were designed the data model and class diagram that fully satisfy the requirements of the web resource.

Key words: architectural style, multi-site, information system, design.

REFERENCES

1. Fowler, M., Rice, D., Fomell, M., Hite, E., Mi, R., Stafford, R. (2006), *Arhitektura korporativnyih programnyih prilozheniy* [Architecture of corporate software applications], Williams, Moscow, Russia.
2. Osadchiy, V., Kruglik, V.S. (2012), *Osnovi rozrobki veb-dodatkov: navchalnyy posibnik* [Fundamentals of web application development. Textbook] LLC "Publishing house MMD", Melitopol, Ukraine.
3. Batiuk, A.E., Mikhailiuk, A.L. (2007), "Automated systems software development of three-level architecture", *Proceedings of the International Conference of Computer Science and Information Technology*, 27 September–29, Lviv, Ukraine, pp. 179–182.
4. Tanenbaum, E., Steen, M. (2003), *Rozpodileni sistemi. Printsipi ta paradigmi* [Distributed Systems. The principles and paradigms], Peter, Sankt-Petersburg, Russia.
5. Sirovych, J., Darui, C. (2008), *Poshukova optimizatsiya saytu (SEO) na PHP dlya profesionaliv. Kerivnitsvo rozrobnika po SEO. Profesiyniy Poshuk* [Search engine optimization (SEO) for PHP professionals. Developer Guide to SEO Professional. Search], Dialectics, Moscow, Russia.
6. Welling, T.L. (2009), *Rozrobka veb-dodatkov na PHP i MySQL* [Web applications in PHP and MySQL], Williams, Moscow, Russia.
7. Rim, A. (2009), *Sovremennyye sistemy upravleniya kontentom* [Modern content management system], Peter, St. Petersburg, Russia.
8. Kartuzov, A.V. (2001), "Using Internet educational re-sources", *Cooperative identity in the new millennium: Abstracts of interuniversity scientific conference*, January 30–31, 2001, Cheboksary, Russia.
9. Vandyuk, J. (2009), *CMS Drupal 6: posibnik z rozrobki sistemi upravlinnya saytom* [CMS Drupal 6: Guide to the development of content management systems], Williams, Moscow, Russia.

Стаття надійшла 15.05.2015.