

Василенко Д. Т., Козонас В. О., Корнійчук М. А.

## ВПРОВАДЖЕННЯ СИСТЕМИ УПРАВЛІННЯ КОНТАКТАМИ ВИПУСКНИКІВ ФАКУЛЬТЕТУ ІНФОРМАТИКИ

У цій статті розглянуто розв'язання проблеми збирання та управління контактами факультету інформатики НаУКМА. У роботі подано огляд архітектури обраного застосунку та описано кроки впровадження CRM системи Odoo. Крім того, продемонстровано інтегрування з цією системою на прикладі порталу випускників.

**Ключові слова:** система управління контактами, CRM система, Odoo, портал випускників.

### Вступ

На факультеті інформатики НаУКМА існувала проблема через брак сукупності стандартизованих контактних даних випускників. Тому метою цієї роботи було створити систему управління контактами факультету інформатики. За основу було взято готове CRM рішення із відкритим кодом – Odoo.

Було поставлено так завдання:

1. Розглянути архітектуру обраної системи та особливості її реалізації.
2. Розгорнути систему та втілити додатковий функціонал.
3. Інтегрувати систему із зовнішнім сервісом порталу випускників.

### Вимоги до системи

Початкові загальні вимоги до системи бути такі:

- 1) простота та швидкість у розгортанні;
- 2) безкоштовне рішення;
- 3) рішення з відкритим кодом для можливості кастомізацій;
- 4) можливість інтеграцій зі сторонніми сервісами.

Джерелом контактів визначено соціальні мережі та контакти, надані різними відділами університету: деканат, відділ по роботі з випускниками тощо.

Серед основних функціональних вимог до роботи із контактами визначено: створення та видалення контакту, імпорт та експорт контактів, можливість архівації, можливість об'єднання у разі дублікатів, редагування.

### Огляд архітектури Odoo системи

Odoo – це потужний набір бізнес-додатків із відкритим кодом, побудованих на основі Open-Object фреймворку. Після першого встановлення Odoo єдиний доступний функціонал – обмежений обмін повідомленнями між користувачами.

Уся додаткова функціональність з'являється за допомогою встановлення модулів відповідно до потреб. Така гнучкість робить Odoo набагато доступнішим порівняно з іншими рішеннями на ринку.

Для налаштування та керування системою Odoo потрібне базове розуміння компонентів, що входять до її складу. Кожен модуль як бізнес-система має набір технологій і програмних платформ для забезпечення його коректного функціонування.

В основі Odoo лежить багатостороння трирівнева архітектура: рівень бази даних для зберігання даних, рівень застосунків та інтерфейсний рівень, що забезпечує користувальницький інтерфейс. Усі вони є окремими шарами в архітектурі.

Рівень застосунків є власне основою, додаткові ж модулі можуть бути встановлені поверх нього. Odoo також наслідує архітектурний патерн MVC – Model View Controller.

На рис. 1 зображено трирівневу архітектуру Odoo.

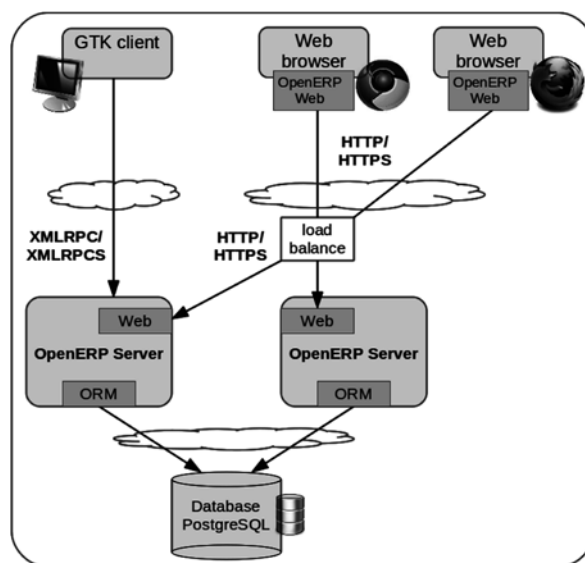


Рис. 1. Архітектура системи Odoo

Як бачимо, система складається із трьох компонентів:

- клієнтська частина як javascript застосунок у браузері;
- Odoо сервер, що містить усю логіку та відповідає за оптимальну роботу застосунку. Одна з частин в Odoо сервері – ORM рушій – призначена для комунікації між сервером та PostgreSQL базою даних, інша – Web – для зв'язку з інтерфейсом;
- PostgreSQL база даних, що містить дані усіх модулів та конфігураційні елементи.

### Розгортання Odoо сервера

Odoо побудований із використанням мови програмування Python та PostgreSQL бази даних для сховища даних – це дві основні вимоги для розгортання Odoо.

Загалом є декілька варіантів встановлення Odoо:

1. Online – онлайн-версія із 15 днями пробного періоду;
2. Packaged installers – інстальтори для Windows та debian-based дистрибутивів, пакети автоматично встановлюють усі залежності;
3. Source install – забезпечує більшу гнучкість, підходить для розробки модулів;
4. Docker – надає базовий docker image.

Щоб встановити Odoо з вихідного коду (source install), потрібно встановити залежні Python бібліотеки. Вихідний код можна завантажити з GitHub.

```
$ git clone https://github.com/odoo/odoo.git
```

Odoо використовує PostgreSQL як систему управління базами даних. За замовчуванням єдиним користувачем є postgres, проте Odoо забороняє під'єднуватися до цього користувача, тому потрібно створити нового.

Рис. 2. Сторінка створення нової бази даних

Для запуску Odoо потрібен Python версії 3.6 або вище. Усі залежності перелічені у файлі requirements.txt, що розташований у корені каталогу. Для встановлення залежностей потрібно ввести: `$ pip install -r requirements.txt`.

Після встановлення всіх залежностей запуснути Odoо можна за допомогою інтерфейсу командного рядка odoo-bin, що розташований у корені каталогу. Типовою командою запуску сервера буде: `python odoo-bin --addons-path=addons`.

За замовчуванням сутності Odoо працюють на порту 8069, тож далі потрібно перейти на `http://<server address>:8069`, щоб отримати доступ до Odoо. Після першого запуску ми побачимо помічника для створення нової бази даних для Odoо (див. рис. 2) [3; 6; 5].

Також Odoо пропонує заповнити базу даних тестовими даними.

Після логіну відкривається сторінка Apps з усіма доступними для встановлення програмами та модулями.

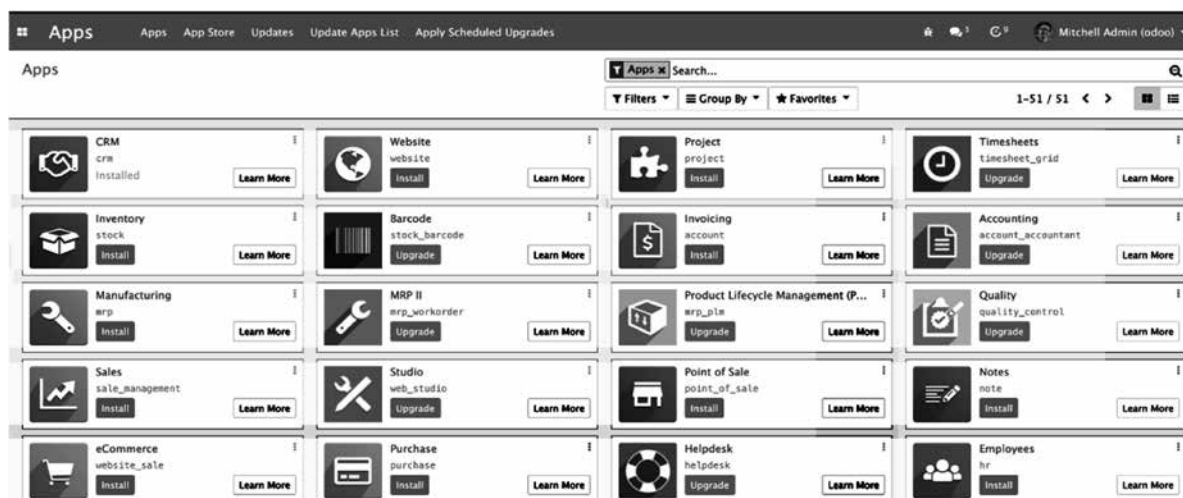
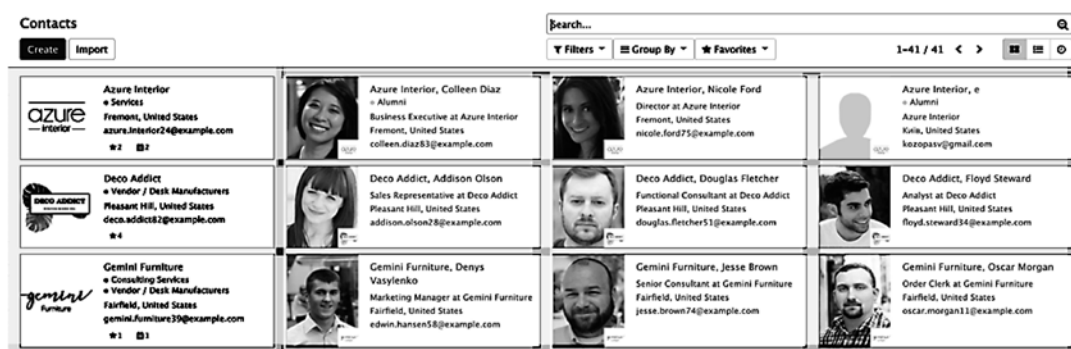


Рис. 3. Модулі Odoо

Рис. 4.  
Модуль CRM



Рис. 5.  
Модуль Contacts



### Кастомізації

Початково встановлена версія Odoo забезпечує лише обмежену систему обміну повідомленнями. Для поповнення функціоналу потрібно встановити додаткові модулі та програми.

Для нашого завдання – системи управління контактами – найкраще підходить модуль CRM, що має також як обов’язкові залежності модулі Contacts та Calendar.

За допомогою модуля Contacts можна легко відслідковувати взаємодії з контактами та зручно впорядковувати інформацію.

Сутності контактів поділяються на два типи: компанія та особа. На кожен із контактів можна навішувати теги, наприклад «Бізнес» або «Випускник», для зручного групування та фільтрації.

У більшості випадків розробка в Odoo означає створення нових модулів. Модулі або додатки (addons) є будівельними блоками для Odoo застосунку. Модуль може додавати нові функції та можливості або ж модифікувати наявні. Власне модуль – це директорія зі файлом маніфесту, `__manifest__.py`, та рештою файлів, що реалізують його особливості [3; 6].

Як правило, вважають поганою практикою модифікувати наявні модулі шляхом зміни вихідного коду напряму. Особливо це стосується офіційних модулів, наданих Odoo. Така практика не дає змоги чітко зробити різницю між початковим кодом і внесеними змінами, що додає складності в оновлення модулів.

Рис. 6.  
Кастомізовані поля



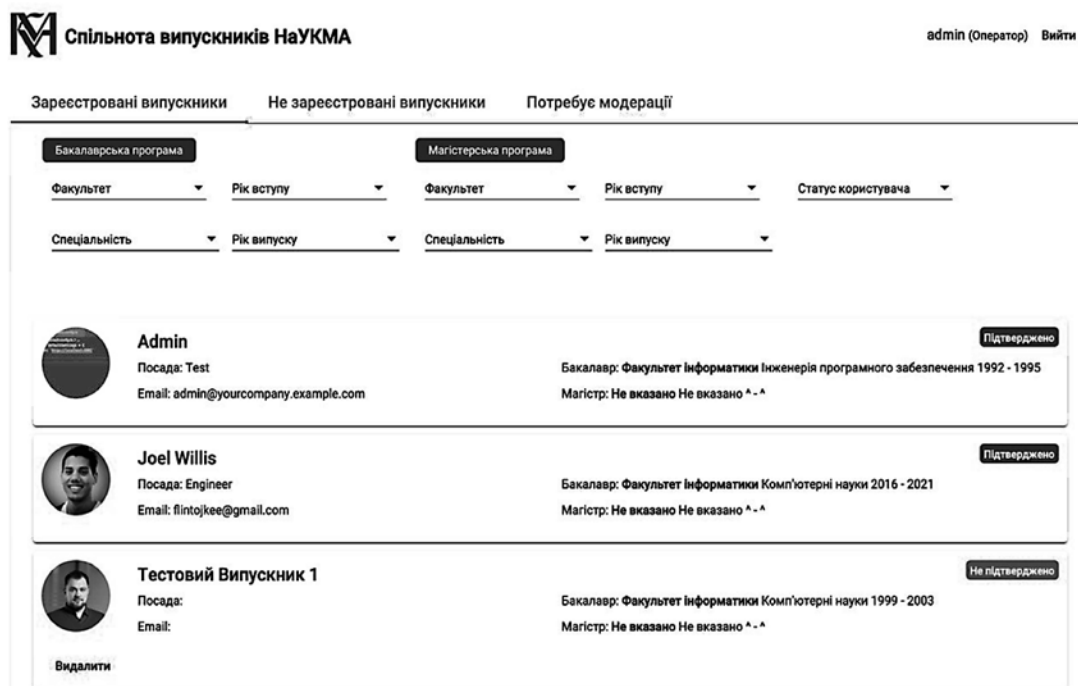


Рис. 7. Портал випускників

Натомість потрібно створити новий модуль – розширення, встановити його поруч із модулем, який хочемо змінити, прописавши в ньому всі необхідні зміни. Кастомізація модуля здійснюється завдяки механізму успадкування, який дає змогу розширювати наявні модулі [3; 6].

Не зважаючи на великий вбудований функціонал, що надає Odoo, все одно виникає потреба додати власні поля відповідно до потреб. На рис. 6 виділено нові поля, додані у власному модулі.

### Інтеграція

Odoo сервер надає зовнішній API, який використовується власним вебклієнтом і також доступний для інших клієнтських додатків. Найпростіший метод доступу до Odoo сервера – це XML-RPC. У Python можна скористатися стандартною бібліотекою `xmlrpc` [1].

Цю систему управління контактами було інтегровано із порталом випускників. Контакти зберігаються на стороні Odoo, їх отримання доступне лише для операторів, а рядові користувачі порталу можуть бачити лише зареєстрованих випускників, тобто таких, що зареєструвались і дали згоду на опублікування своїх даних. На рис. 7 зображено інтерфейс порталу, де відображаються дані із Odoo.

### Висновки

У процесі роботи було досліджено проблему збирання та управління контактами у межах факультету інформатики НаУКМА. Врешті було реалізовано систему для збирання та управління контактами на основі Odoo сервера. Нині систему запущено на сервері факультету й інтегровано з порталом випускників.

### Список літератури

1. External API [Electronic resource]. – Mode of access: <https://www.odoo.com/documentation/13.0/webservices/odoo.html>.
2. Installing Odoo [Electronic resource]. – Mode of access: <https://www.odoo.com/documentation/13.0/setup/install.html#setup-install-source>.
3. Moss G. Working with Odoo / Greg Moss. – 2015. – 399 p.
4. Odoo Architecture [Electronic resource]. – Mode of access: [https://doc.odoo.com/trunk/server/02\\_architecture](https://doc.odoo.com/trunk/server/02_architecture).
5. Odoo CRM Features [Electronic resource]. – Mode of access: <https://www.cybrosys.com/blog/odoo-crm-features>.
6. Reis D. Odoo 10 development essentials / Daniel Reis. – 2016. – 273 p.

### References

- Moss, G. (2015). *Working with Odoo: learn how to use Odoo, a resourceful, open source business application platform designed to transform and modernize your business*. Birmingham: Packt Publ.
- Odoo. (n.d.). Bushfire resources: External API. Retrieved from <https://www.odoo.com/documentation/13.0/webservices/odoo.html>.

Odoo. (n.d.). Bushfire resources: Installing Odoo. Retrieved from <https://www.odoo.com/documentation/13.0/setup/install.html#setup-install-source>.

Odoo. (n.d.). Bushfire resources: Odoo Architecture. Retrieved from [https://doc.odoo.com/trunk/server/02\\_architecture](https://doc.odoo.com/trunk/server/02_architecture).

Odoo. (n.d.) Bushfire resources: Odoo CRM Features. Retrieved from <https://www.cybrosys.com/blog/odoo-crm-features>.

Reis, D. (2016). *Odoo 10 development essentials: Fast-track your development skills to build powerful Odoo 10 business applications*. Birmingham, UK: Packt Publishing.

D. Vasylenko, V. Kozopas, M. Korniiichuk

## IMPLEMENTATION OF THE CONTACT MANAGEMENT SYSTEM OF GRADUATES OF THE FACULTY OF INFORMATICS

*This article considers the solution to the problem of collecting and managing contacts of the Faculty of Informatics of NaUKMA. The problem statements were as follows: consider the architecture of the selected system and features of its implementation, deploy the system, and implement additional functionality, integrate with the external service of the alumni portal. Based on the above problem, the purpose of this work was set to the creation of a contact management system of the Faculty of Informatics. In particular, it was decided to use an available open-source CRM solution as a basis. This method allows us to quickly get a tool with the main functionality built-in for further development and implementation of customization. The paper cover information about the ways to deploy an Odoo server, the procedure for creating its own module for customization and changes, and also gives an example of using the External API for the external integrations. As a result, this application has been integrated with the alumni service.*

*At the Faculty of Informatics of NaUKMA there was a problem of lack of a set of standardized contact details of graduates. Based on the above problem, the purpose of this work was to create a contact management system of the Faculty of Informatics. In particular, the ready-made CRM solution with open source – Odoo was taken as a basis.*

*The problem statement is as follows: consider the architecture of the selected system and features of its implementation; deploy the system and implement additional functionality; integration with the external service of the alumni portal.*

*The following items represent the initial general system requirements: simplicity and speed in deployment; free solution; open source solutions for customization; the possibility of integration with third-party services.*

*The source of contacts is social networks and contacts provided by various departments of the university: the dean's office, the department for work with graduates, etc.*

*Among the main functional requirements for working with contacts are: creating and deleting a contact, import and export of contacts, the ability to archive, the ability to merge in the case of duplicates, editing.*

**Keywords:** contact management system, CRM system, Odoo, alumni portal.



Creative Commons Attribution 4.0 International License (CC BY 4.0)

Матеріал надійшов 10.06.2020