

УДК 004.4

РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ АНАЛІЗУ, СПОВІЩЕННЯ ТА ОБРОБКИ ЧЕРГ НА КАСАХ В МАГАЗИНАХ

Бондар Владислав Русланович

к.т.н., доцент каф ІС Кузнецов Микола Олександрович
Національний університет «Одеська політехніка», УКРАЇНА

АНОТАЦІЯ. У роботі розглянуто процес створення інформаційної системи для діджиталізації процесів управління чергами на касах магазинів, у тому числі: сповіщення персоналу магазину та управління про виявлення черги на конкретній касі, статистика щодо черг на магазинах, аналіз тривалості черги та пікових моментів навантаження на магазині.

Вступ. У сучасному світі, важко уявити життя без походу у крамницю і кожного разу ми відстоюємо у черзі. У середньому людина за своє життя відстоює близько 1,1 року у чергах, тому було б дуже добре намагатися зменшити цю величину.

Сьогодні розробляють різноманітні прийоми для поліпшення рівня обслуговування клієнтів [1]. Але такі алгоритми найчастіше мають недоліки у вигляді людського фактору, тому розробка інформаційної системи з аналізу, сповіщення та обробки черг на касах в магазинах є досить актуальною.

Мета роботи. Метою роботи є розробка інформаційної системи з аналізу, сповіщення та обробки черг на касах в магазинах на основі клієнт-серверної архітектури.

Основна частина роботи. Для досягнення поставленої мети передбачено наступні кроки: аналіз аналогічних систем управління чергами для виявлення функціональності, яку варто перейняти; вибір інструментів для проектування; розробка функціональних вимог, макетів, діаграм шарів, процесів, даних, комунікацій, класів та допоміжних діаграм для розробки інформаційної системи; реалізація самої системи; тестування функціоналу розробленої інформаційної системи.

При огляді існуючих систем, зокрема *QLess*, *Wavetec* та *Qminder*, було виявлено їх унікальний функціонал.

QLess[2] пропонує хмарну платформу, яка дозволяє клієнтам резервувати час прийому та сповіщати їх про стан черги. Це зручно для клієнтів, оскільки вони можуть уникнути витрати часу на очікування у черзі, зарезервувавши заздалегідь час, коли вони хотіли би отримати обслуговування. Крім того, сповіщення про стан черги дозволяє клієнтам бути усвідомленими щодо того, коли їх черга наближається.

Wavetec [3] спеціалізується на розробці технологій управління чергами та цифрових рішень для бізнесу. Їх системи дозволяють ефективно керувати чергами, забезпечуючи оптимальний розподіл ресурсів та мінімізуючи час очікування клієнтів. Такі рішення особливо корисні для підприємств з великим потоком клієнтів, де важливо ефективно організувати процес обслуговування.

Qminder [4] є хмарним сервісом для управління чергами та підвищення якості обслуговування клієнтів. Вони пропонують функціонал, який дозволяє підприємствам створювати інтерактивні табло, відстежувати час очікування та керувати потоком клієнтів. Це може покращити задоволеність клієнтів, оскільки вони будуть більш інформовані про стан черги та можуть отримати швидше обслуговування.

Порівняння систем, можна побачити у таблиці 1, в якій вказані їхні переваги, функціонал та можливості. Під час порівняння різних систем, яке було проведено для даної роботи, виявлено, що розроблювана система матиме базовий функціонал, який є стандартним для подібних застосунків. Проте, система даної роботи буде мати додаткові можливості, які дозволять сповіщати персонал на магазині через телеграм бота, надавати повну статистику щодо черг у торговій мережі, також буде адмін панель для налаштування системи.

Таблиця 1 – Порівняльні характеристики офісних пакетів

Функціональна можливість	Назва системи			
	<i>Qless</i>	<i>Wavetec</i>	<i>Qminder</i>	Розроблюєма система
Відстеження кількості людей у черзі	+	+	+	+
Сповіщення персоналу через телеграм бот	-	-	-	+
Показання часу очікування у черзі	+	+	-	+
Відображення статистики черг на кожній касі магазинах	-	+	+	+
Адмінпанель для додавання нових магазинів та користувачів	+	-	-	+
Сайт з різноманітною статистикою по магазинам та касам на них	+	-	-	+

При виборі архітектурного шаблону для розробки системи було розглянуто декілька варіантів. Один з них - трирівнева архітектура [5], яка складається з наступних компонентів: представлення даних, прикладний компонент і керування ресурсами. В такій архітектурі декілька серверів обробляють запити користувачів, розподіляючи навантаження і забезпечуючи оптимізацію ресурсів. Також вкладено можливість розширення трирівневої архітектури до багаторівневої (*N-tier*, *Multi-tier*) шляхом додавання додаткових серверів. Багаторівнева архітектура дозволить підвищити ефективність роботи системи і оптимізувати розподіл програмно-апаратних ресурсів.

В результаті аналізу потреб системи і специфіки взаємодії з відеообладнанням, було вирішено обрати клієнт-серверний архітектурний стиль. Цей стиль дозволяє ефективно розділити функціонал і обчислювальне навантаження між клієнтськими додатками (замовниками послуг) і серверними додатками (постачальниками послуг). Клієнти відправляють запити до сервера, де вони обробляються, і результати повертаються назад до клієнтів. Завдяки цьому архітектурному стилю сервер може обслуговувати декілька клієнтів одночасно, оптимізуючи час відповіді і навантаження системи.

Для розробки *back-end* частини додатку було обрано мову програмування *Python*. Вибір *Python* обумовований кількома факторами.

По-перше, *Python* є однією з найпопулярніших мов програмування, що дозволяє мати доступ до широкого спектру ресурсів, бібліотек та інструментів для розробки. Це робить його дуже зручним і потужним інструментом для створення складних систем.

По-друге, *Python* має простий і зрозумілий синтаксис, що сприяє швидкому розробленню і підтримці коду. Це особливо важливо, оскільки планується розширювати та підтримувати систему у майбутньому.

Крім того, *Python* має багато спеціалізованих бібліотек і фреймворків для розробки веб-додатків, обробки даних та машинного навчання. Планується використовувати деяких з цих інструментів для реалізації функцій системи, що дозволить зосередитись на вирішенні конкретних завдань замість написання власного коду з нуля. З урахуванням всіх цих факторів, вибір *Python* як мови програмування для розробки *back-end* частини даної системи є раціональним і обґрунтованим рішенням.

Для розробки *front-end* частини буде використана мова *JavaScript*. Через її гнучкість, універсальність, та наявні фреймворки, що дозволяють швидко та елегантно реалізувати клієнтський інтерфейс та взаємодію з *back-end* сервісами.

Обираючи СУБД для веб-додатка, варто врахувати кілька основних моментів, спираючись на масштаб програми, обсяг даних, швидкість і надійність. *PostgreSQL* і *MySQL* є двома

найпопулярнішими системами управління базами даних з відкритим вихідним кодом. Та хоч обидві СУБД мають свої вагомні переваги та певні обмеження, *PostgreSQL* [6] зазвичай вважається більш передовим вибором для складніших та масштабних додатків.

Ось декілька міркувань, чому *PostgreSQL* може бути кращим вибором, порівняно з *MySQL*:

1. Надійність та цілісність даних: *PostgreSQL* гарантує високий рівень надійності та цілісності даних, завдяки застосуванню транзакцій та контролю цілісності даних.

2. Підтримка геоданих: *PostgreSQL* має вбудовану підтримку для опрацювання геоданих, що спрощує роботу з географічними даними та здійснення запитів на основі геолокації.

3. Підтримка *JSON* та *JSONB*: *PostgreSQL* володіє вбудованою підтримкою *JSON* та *JSONB*, що дає змогу зберігати та обробляти структуровані дані у форматі *JSON*.

4. Висока продуктивність: *PostgreSQL* зазвичай демонструє кращу продуктивність, порівняно з *MySQL*, коли стикається з виконанням складних запитів та обробкою великої кількості одночасних підключень.

Загалом, *PostgreSQL* здатний забезпечити високу надійність та масштабованість, що робить його ідеальним вбтві з вимогами складних та масштабних веб-додатків. Його стабільність та здатність ефективно працювати з великими обсягами даних забезпечують безперебійну роботу навіть при великому навантаженні.

У контексті проектування інформаційної системи був обраний патерн *Model-view-controller (MVC)*, оскільки він сприяє створенню ефективного, розширюваного та легкого для тестування та обслуговування додатку. Використання *MVC* також дозволяє розробникам зручно працювати над проектом та скоротити час, необхідний для розробки та впровадження нової функціональності.

Висновки. Було проведено огляд трьох систем з подібним функціоналом, визначено їх переваги та недоліки. З урахуванням результатів аналізу, планується розробити конкурентоспроможну систему, яка уникне недоліків попередніх систем та використовуватиме їх переваги. Для розробки обрано архітектуру "Клієнт-Сервер" та мову програмування *Python*, яка має розвинені та зручні у використанні фреймворки та бібліотеки для розробки. Система управління базами даних *PostgreSQL* була обрана для зберігання даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Artemenko A., Mehrez I., Govindaraj K., Kirstaedter A., Kuznietsov M. Empirical investigation of offloading decision making in industrial edge computing scenarios. 2021 Joint European Conference on Networks and Communications and 6G Summit, EuCNC/6G Summit 2021, 2021, P. 311–316, 9482556
2. Qless[Електроннийресурс].-Режимдоступу:<https://qless.com/>
3. Wavetec[Електроннийресурс].-Режимдоступу:<https://www.wavetec.com/>
4. Qminder[Електроннийресурс].-Режимдоступу:<https://www.qminder.com/>
5. Трирівневаархітектура[Електроннийресурс]. - Режимдоступу:<https://training.qatestlab.com/blog/technical-articles/client-server-architecture/>
6. PostgreSQL[Електроннийресурс].-Режимдоступу:<https://www.postgresql.org/>

DEVELOPMENT OF THE INFORMATION SYSTEM OF ANALYSIS, NOTIFICATION AND PROCESSING OF QUEUES AT THE CASHIER IN STORES

Vladyslav Bondar

Ph.D., Associate Professor of CISN Mykola Kuznietsov
Odesa Polytechnic National University, UKRAINE

ANNOTATION. The paper examines the process of creating an information system for the digitization of queue management processes at store cash registers, including: notification of store staff and management about the detection of a queue at a specific cash register, statistics on queues at stores, analysis of queue duration and peak load moments in the store.