

УДК 004.4

Одарущенко О. Б., к.т.н., доцент,
Івахнов Б. О., магістрант
Полтавський національний технічний
університет імені Юрія Кондратюка

РОЗРОБКА СИСТЕМИ ОБЛІКУ ДІЯЛЬНОСТІ ЗАКЛАДУ РЕСТОРАННОГО ТИПУ

Анотація. У статті розглянуті методи розробки сервісу управління діяльністю закладу ресторанного типу. Мета роботи полягає в розробці сервісу управління діяльністю закладу ресторанного типу із обмеженим обсягом функцій, що дозволить ефективно та зручно проводити щоденні операції користувачем. Методи розробки включають вивчення матеріалу, використання об'єктно-орієнтованого програмування, робота з базами даних MySQL.

Ключові слова: управління діяльністю закладу ресторанного типу, PHP, Yii2, ООП, СУБД MySQL.

Вступ

Основою роботи закладу є взаємодія з відвідувачами. Крім основного завдання, надання послуг відвідувачам, закладу потрібно вирішити ряд таких задач, як ведення обліку надходження товару, ведення списку замовлень, проведення інвентаризації, створення звітів про продаж товару. Вище наведений список є далеко не повним переліком завдань, які потрібно автоматизувати для ефективної роботи закладу.

Для спрощення роботи закладу існує ряд засобів автоматизації. До них відносяться спеціалізовані системи, які відрізняються кількістю додаткових

функцій, але важкі в попередньому налаштуванні. На даний момент доступний великий перелік спеціалізованих систем обліку, які призначенні закладів різних розмірів та масштабів. Найбільш поширеними на території України є програмні продукти R-Keeper [1], Poster [2]. Дані системи дозволяють вести облік приходу товару, список замовлень, проводити автоматизацію та показують звіти про продаж товару, але кожна з них є дорогою та не всі системи обслуговуються постійно.

Таким чином, *актуальним завданням*, є розробка програми, яка дозволить автоматизувати роботу закладу та підвищить ефективність роботи персоналу і при цьому буде містити необхідну кількість функціоналу, яка щоденно потрібна для користувача. Зменшення кількості функцій зменшить складність первинного налаштування та подальшого використання системи.

Виходячи з актуальності поставленої задачі, *метою роботи* є розробка системи обліку закладу ресторанного типу із обмеженим обсягом функцій, що дозволить ефективно та зручно проводити щоденні операції.

Для досягнення поставленої мети необхідно вирішити низку *окремих задач*, а саме проаналізувати існуючі системи обліку для закладів ресторанного типу, сформулювати технічне завдання на програмний продукт, що розробляється, розробити логіку функціонування програмного продукту, розробити інтерфейс програмного продукту, спроектувати та розробити базу даних для зберігання необхідної інформації, щодо програмного продукту, спроектувати та розробити програмний продукт, провести верифікацію та тестування програмного продукту, його інтерфейсу та бази даних.

Розроблений у *результаті* додаток до спеціалізованих систем обліку має дозволяти у напівавтоматичному режимі здійснювати контроль за визначеними пунктами діяльності закладу.

Для розробки веб-додатку буде використано фреймворк Yii2, мову програмування PHP. Даний вибір обумовлений простотою опанування технології, оскільки фреймворк Yii2 надає вичерпаний набір документації.

Необхідний та достатній перелік функцій розроблюваного сервісу

Додаток, що створюється повинен надавати можливість управління складами, замовленнями, столам та перегляд звітності і залишку товару на складах(інвентаризація), а також забезпечити можливість формування меню з урахуванням товару, який внесений на склади.

В частині управління складами, користувач повинен мати змогу додавати, видаляти та редагувати опис складу. Також користувач має додавати, редагувати або видаляти існуючий товар зі складу.

В частині управління замовленнями користувач повинен мати змогу додавати, редагувати, закривати та видаляти замовлення. Також користувач має додавати, редагувати, закривати й видаляти замовлення, з урахуванням товарів із меню.

Звіт включає в себе інформацію про продаж товарів, з можливістю вибрати фільтри по даті та по товару із меню.

Інвентаризація містить інформацію про залишки товарів на складах.

1. Архітектура розроблюваного сервісу

Багаторівнева архітектура є однією з найбільш використовуваних моделей архітектури в галузі інформаційних технологій. Вона використовується так часто, тому що масштабується, розширюється, безпечна і яку підлягає обслуговуванню згодом. Це також допомагає розробникам із різними ролями краще виконувати свою роботу, не заважаючи один одному.

На Рис. 1 наведено багаторівневу архітектуру додатку.

Рівень, що стоїть вище за ієрархією, користується представленими нижче рівнями, але не інформований про наявність сусіднього верхнього рівня. Більш того, зазвичай кожен проміжний рівень приховує нижній рівень від верхнього.

Переваги розбиття системи на рівні:

- окремий рівень можна сприймати як єдине самодостатнє ціле;

- можна вибрати альтернативну реалізацію базових рівнів;
- залежність між рівнями можна звести до мінімуму.

Недоліки:

- рівні здатні вдало інкапсулювати багато, але не все: модифікація одного рівня пов'язана з необхідністю внесення каскадних змін в інші рівні;
- наявність надлишкових рівнів нерідко знижує продуктивність системи.

Короткий опис для кожного рівня:

1. Рівень представлення – відповідає за розміщення призначеного для користувача інтерфейсу додатку. Наприклад, в цьому рівні будуть всі призначені для користувача елементи .aspx, .ascx або .cshtml разом з файлами JavaScript і CSS. Методи, які відповідають за отримання даних з бази даних, будуть посилатися тільки безпосередньо з рівня бізнес-компонентів або з рівня сервісів. На цьому рівні можна використовувати такі технології, як PHP pattern MVC.

2. Рівень сервісів – можливе подання рівня бізнес-компонентів як API для сторонніх систем. Наприклад, якщо потрібно поділитися з мобільним додатком деякими даними про продукти, які є в базі даних, цей рівень має нести за це відповідальність. Однак цей рівень не є обов'язковим. Методи, які необхідно відобразити в таких межах, посилаються на рівень бізнес-компонентів, і цей рівень повинен використовуватися тільки рівнем представлення і іншими сторонніми системами.

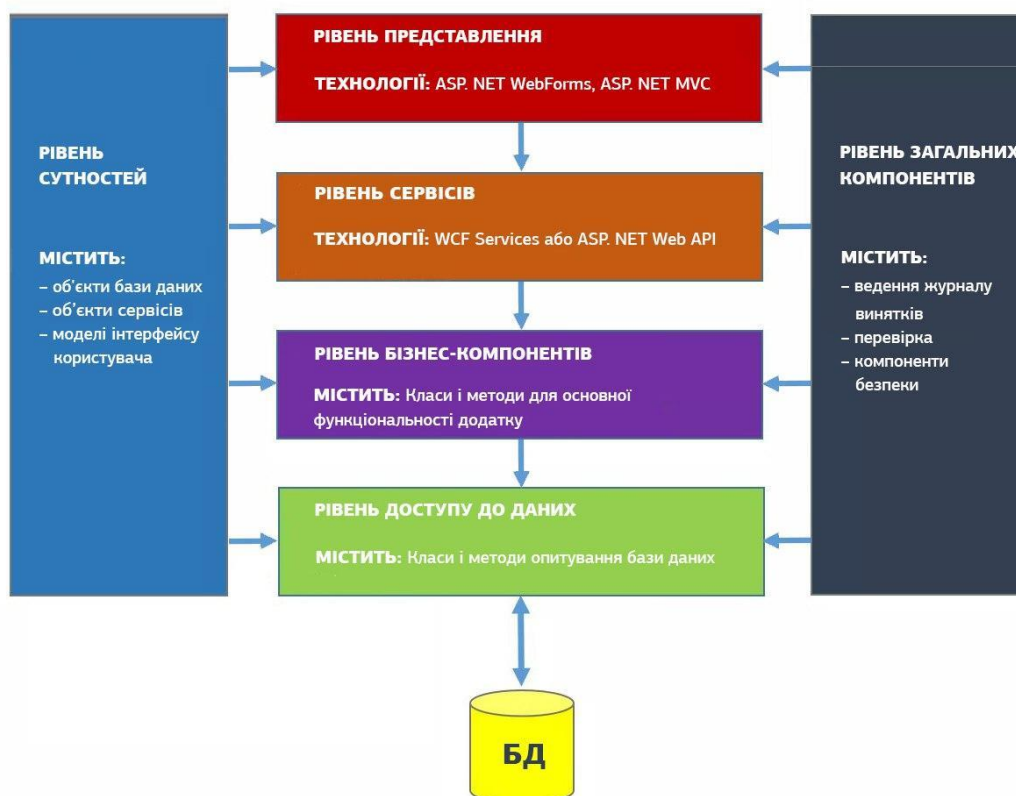


Рис. 1. Багаторівнева архітектура додатку

3. Рівень бізнес-компонентів – містить всі основні функції програми. Метою є утримувати всю логіку, яка застосовується до методів, які відображаються з рівня доступу до даних, до того, як вони будуть вказані в рівні уявлення або на рівні служб. Такі операції, як перетворення масивів в списки, математичні обчислення або перетворення змінних, виконуються на цьому рівні. Об'єкти, необхідні для зазначених вище операцій, будуть посилатися з рівня сутностей, а на методи роботи з базою даних будуть посилатися з рівня доступу до даних.

4. Рівень доступу до даних – містить функціональні можливості для створення, повернення, поновлення, видалення (CRUD) елементів в базі даних. На цьому рівні можна використовувати такі технології, як Yii2 Gi. Об'єкти або класи зіставлення таблиць бази даних, необхідні для зазначених вище операцій, будуть посилатися із рівня сутності. Цей рівень повинен використовуватися тільки на рівні бізнес-компонентів.

5. Рівень сутностей – містить усі сутності, які використовуються в усіх інших проектах програми, такі як співставлення таблиць з бази даних, DTO (об'єкти передачі даних) або Yii2 MVC моделі.

6. Рівень загальних компонентів - містить усі загальні бібліотеки або функціональні можливості, які можуть використовуватися на будь-якому із зазначених вище рівнів. Наприклад, на цей рівень можуть бути додані функції перевірки, бібліотеки безпеки, інструменти шифрування або класи реєстрації винятків. Це необов'язковий рівень, і не обов'язково мати будь-які залежності від вищеописаних рівнів, тому що цей рівень може бути скопільований і розділений між кількома проектами.

Інтерфейс користувача

Для спрощення і прискорення процесу розробки інтерфейсу користувача використовується CSS фреймворк Bootstrap.

Далі буде показаний зовнішній вид розробленого інтерфейсу.

На Рис. 2 – 3 зображено сторінки автоматизації та списку замовлень.

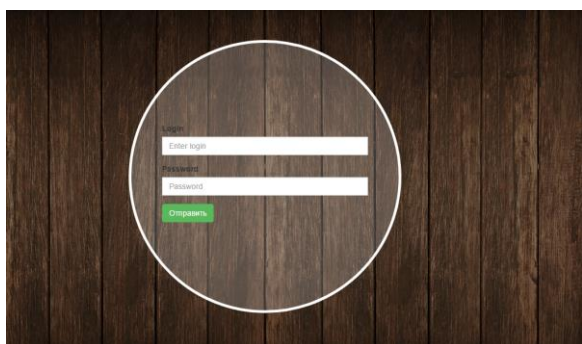


Рис. 2. Сторінка авторизації

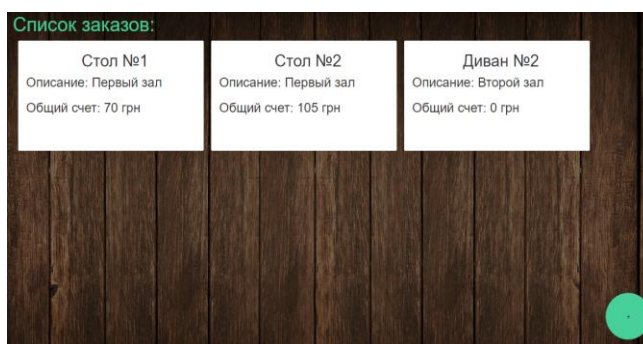


Рис. 3. Сторінка зі списком замовлень

Логічна модель бази даних

Модель сутність-зв'язок (ER-модель) (англ. Entity-relationship model, ERM) - модель даних, що дозволяє описувати концептуальні схеми предметної області. ER-модель використовується при високорівневої (концептуальному)

проектуванні баз даних. З її допомогою можна виділити ключові сутності і позначити зв'язки, які можуть встановлюватися між цими сутностями.

ER-модель являє собою формальну конструкцію, яка сама по собі не наказує ніяких графічних засобів її візуалізації. Як стандартна графічної нотації, за допомогою якої можна візуалізувати ER-модель, була запропонована діаграма сутність-зв'язок (ER-діаграма). Кожен користувач має унікальний номер, ім'я, пароль та емейл. Сутність «Користувач» має роль. Склад може мати декілька або багато товарів. Товари, в свою чергу входять до пунктів меню, кожен пункт меню може складатися з декількох товарів. Пункти меню додаються до замовлення, яке співробітник може додати до столу. Замовлення оплачується в касі.

На Рис. 4 наведено логічну модель бази даних.

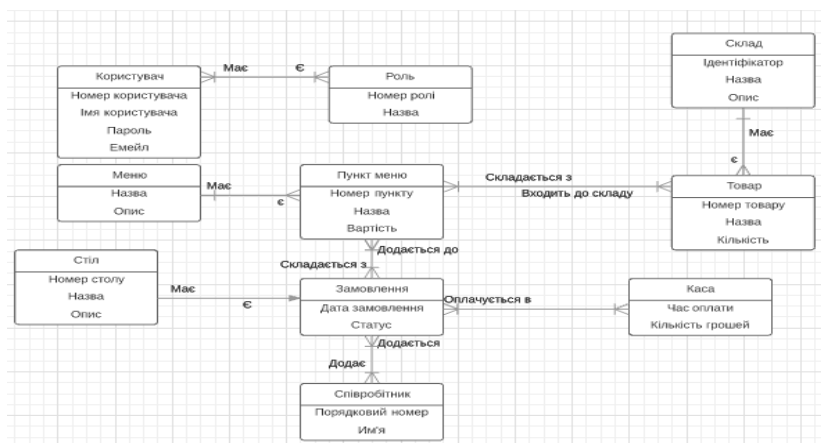


Рис. 4. Логічна модель бази даних

Тестування

При розробці програмного забезпечення в якості методології тестування був обраний підхід TDD (Test Driven Development). Даний підхід дозволяє спочатку підготувати тести для окремих компонентів системи і "підігнати" реалізацію під вже існуючі тести. Таким чином не потрібно витратити час на написання тестів після розробки програми.

Для написання тестів була вибрана модель AAA (Arrange, Act, Assert):

Розділ Arrange ініціалізує об'єкти і встановлює значення даних, котрі передаються до метода, який буде протестовано.

Розділ Act, викликає метод, який потрібно протестувати та передає йому параметри із Arrange, якщо такі існують.

Розділ Assert перевіряє, що результат дії методу, який тестується буде такий, як очікувалось.

Для порівняння очікуваних значень та реальних використовується клас CollectionAssert та клас Assert і його метод AreEqual Для завідомо помилкових тестів, в яких повинні генеруватися виключення, використовується атрибут ExpectedException.

Висновки

Відповідно до поставленої задачі, а саме розробка системи обліку закладу ресторанного типу із обмеженим обсягом функцій, що дозволить ефективно та зручно проводити щоденні операції, було розроблено веб-додаток для обліку діяльності закладу ресторанного типу. При аналізі предметної області були досліджені аналогічні системи з більш розширеним функціоналом. Прийнято рішення введення у створювану систему таких наступних щоденно необхідних дій, як управляти складами, замовленнями, столами, створювати звіти про продаж товарів та їх залишок на складах.

На етапі проектування обрано архітектуру проекту, а саме багаторівневу архітектуру. Спроектовано логічну модель бази даних, на основі якої були виділені основні сутності фізичної моделі бази даних, створено діаграму варіантів використання, що показує можливий функціонал для двох типів користувачів.

На етапі розробки була реалізована архітектура додатку, на основі сутностей фізичної моделі була створена діаграма бази даних.

Розроблений сервіс задовольняє всім вимогам, сформульованим під час постановки задачі.

Даний проект буде вдосконалений та доповнений новими функціональними можливостями, такими як перенесення та розбиття замовлень, мобільний додаток. На даний момент веб-додаток було розроблено під певний заклад із врахуванням всіх практичних задач замовника.

Посилання

1. *R-Keeper. Офіційний сайт [Ел. ресурс] – Режим доступу: <http://rkeeperua.com>*
2. *Poster. Офіційний сайт [Ел. ресурс] – Режим доступу: <http://joinposter.com/ua>*
3. *Дейт К. Дж. Введение в системы баз данных. Introduction to Database Systems. — 8-е изд. — М.: Вильямс, 2005. — 1328 с.*

Authors:

Olena Odarushenko, Ivakhnov Bohdan

DEVELOPMENT OF THE ACCOUNTING SYSTEM OF ACTIVITIES RESTAURANT TYPE

Abstract. In the article reviewed the methods Development of the accounting system of activities restaurant type. The purpose of the work is to develop a document management system for a company with a limited scope of functions, which will allow the user to conduct daily operations efficiently and conveniently. Development methods include studying the material, using object-oriented programming, working with databases MySQL.

Keywords: accounting system of activities restaurant type, OOP, PHP, Yii2, MySQL.

Авторы:

Е. Б. Одарущенко, Б. О. Ивахнов

РАЗРАБОТКА СИСТЕМЫ УЧЕТА ДЕЯТЕЛЬНОСТИ ЗАВЕДЕНИЯ РЕСТОРАННОГО ТИПА

Аннотация. В статье рассмотрены методы разработки сервиса управления деятельностью заведения ресторанного типа. Цель работы заключается в разработке сервиса управления деятельностью заведения ресторанного типа с ограниченным объемом функций, что позволит эффективно и удобно проводить ежедневные операции пользователем. Методы разработки включают изучение материала, использование объектно-ориентированного программирования, работа с базами данных MySQL.

Ключевые слова: управление деятельностью заведения ресторанного типа, PHP, Yii2, ООП, СУБД MySQL.