

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

ДОПУСТИТИ ДО ЗАХИСТУ  
Завідувач кафедри

\_\_\_\_\_Литвиненко О.Є.

«\_\_\_» \_\_\_\_\_ 2021 р.

**ДИПЛОМНИЙ ПРОЕКТ**  
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

**ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ  
"БАКАЛАВР"**

Тема: \_\_\_\_\_ «Програмний модуль системи онлайн замовлень в ресторані»  
\_\_\_\_\_

Виконавець: \_\_\_\_\_ Чижик О. В.

Керівник: \_\_\_\_\_ Нечипорук О.П.

Нормоконтролер: \_\_\_\_\_ Тупота Є.В.

**Київ 2021**

# НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет \_\_\_\_\_ кібербезпеки, комп'ютерної та програмної інженерії

Кафедра \_\_\_\_\_ комп'ютеризованих систем управління

Освітнього ступеня \_\_\_\_\_ бакалавр

Спеціальність \_\_\_\_\_ 123 "Комп'ютерна інженерія"  
(шифр, найменування)

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Литвиненко О. Є.

« \_\_\_\_\_ » \_\_\_\_\_ 2021 р.

## ЗАВДАННЯ на виконання дипломного проекту

\_\_\_\_\_ Чижика Олега Володимировича

(прізвище, ім'я, по батькові)

**1. Тема роботи:** "Програмний модуль системи онлайн замовлень в ресторані"

затверджена наказом ректора від "04" \_\_\_\_\_ лютого 2021 року № 135 /ст.

**2. Термін виконання роботи:** з 17.05.2021 до 20.06.2021

**3. Вихідні дані до роботи:** системна програмна документація,  
документація на операційні системи

**4. Зміст пояснювальної записки (перелік питань, що підлягають розробці):**

1) аналіз предметної області;

2) аналіз вимог до програмного забезпечення;

3) реалізація захисту клієнт-серверної архітектури

**5. Перелік обов'язкового графічного матеріалу:**

1) діаграма послідовностей методу отримання даних з серверу;

2) діаграма архітектури клієнт-серверної системи;

3) сторінка входу в систему онлайн замовлень в ресторані;

## 6. Календарний план

№ п/п	Етапи виконання дипломного проекту	Термін виконання етапів	Примітка
1	Ознайомитись з постановкою задачі дипломного проекту	17.05.21-19.05.21	
2	Вивчити спеціальну літературу і технічну документацію	19.05.21-20.05.21	
3	Проаналізувати принципи збереження інформації у різних файлових системах	21.05.21-24.05.21	
4	Написати розділи дипломного проекту щодо принципів збереження інформації у різних файлових системах	25.05.21-31.05.21	
5	Проаналізувати алгоритми для програмної реалізації задачі моніторингу системних параметрів файлової організації	01.06.21-04.06.21	
6	Написати розділ дипломного проекту щодо програмної реалізації	05.06.21-15.06.21	
7	Підготувати графічний демонстраційний матеріал, Оформити відгук і рецензію	15.06.21-20.06.21	

## 7. Дата видачі завдання « 17 » травня 2021 р.

Керівник дипломного проекту \_\_\_\_\_ Нечипорук О.П.  
(підпис)

Завдання прийняв до виконання \_\_\_\_\_ Чижик О.В.  
(підпис студента)

## РЕФЕРАТ

Пояснювальна записка до дипломного проекту “Програмний модуль системи онлайн замовлень в ресторані ”: основна частина – 40 с., 5 рис., 2 табл., 15 літературних джерел.

ОНЛАЙН ЗАМОВЛЕННЯ, ІНФОРМАЦІЙНА СИСТЕМА, КОНТРОЛЬ ЯКОСТІ, ФАЙЛ, КЛІЄНТ, АДМІНІСТРАТОР.

**Об’єкт дослідження** – програмний модуль системи онлайн замовлень в ресторані.

**Предмет дослідження** – інформаційна система онлайн замовлень в ресторані.

**Мета дипломного проекту** – проектування та створення програмного модуля системи онлайн замовлень в ресторані

**Метод проектування** – визначення основних принципів роботи системи замовлень в ресторані, розробка програмного забезпечення з застосуванням мов програмування високого рівня.

Прогнозні припущення щодо розвитку об’єкта дослідження – впровадження програмного модуля для безконтактної оплати.

Результати дипломного проекту рекомендується використовувати при розробці програмних засобів, призначених для систем ресторанного бізнесу.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ.....	6
ВСТУП .....	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
1.1. Сучасні технології в задачах удосконалення процесів онлайн-замовлень з ресторанів.....	11
1.2. Оптимізація бізнес процесів на основі впровадження інформаційного додатку .....	14
1.3. Опис існуючих систем онлайн замовлень .....	16
1.4. Висновки до розділу .....	20
РОЗДІЛ 2 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	21
2.1. Функціональні вимоги.....	21
2.2. Нефункціональні вимоги.....	24
2.3. Висновки до розділу .....	26
РОЗДІЛ 3 СТРУКТУРА СИСТЕМИ ОНЛАЙН ЗАМОВЛЕНЬ В РЕСТОРАНІ ...	27
3.1. Серверна частина системи .....	27
3.2. База даних .....	30
3.3. Діаграма класів .....	33
3.4. Висновки до розділу .....	36
ВИСНОВКИ.....	37
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ.....	39
ДОДАТОК А .....	41
ДОДАТОК Б.....	42

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

- ІС – інформаційна система
- MVC – *Model View Controller*
- SOAP – *Simple Object Access Protocol*
- GUI – *graphical user interface*
- JSON – *JavaScript Object Notation*
- HTTP – *Hyper Text Transfer Protocol*
- ПП – програмний продукт
- СКБД – система контролю баз даних

## ВСТУП

У наш час у багатьох ресторанах все ще використовують традиційний спосіб замовлення послуг, оскільки ми бачимо, що офіціант використовує ручку та папір, коли приймає замовлення клієнта. Це метод з низькою ефективністю, незручний і може містити помилки. Наприклад, офіціант втратив папір для замовлення у метушні або в іншій ситуації, почерк офіціанта важко зрозуміти іншим людям, що може призвести до того, що кухня та порціонер псують замовлення, а також можуть спричинити помилки в розрахунках. Ця ситуація, якщо це трапляється часто, може призвести до того, що клієнту не сподобається сервіс ресторану і він не захоче приїхати знову, і це спричинить великий вплив на ресторан. Використовуючи систему замовлення ресторанів за допомогою мобільних додатків, це робить систему замовлення більш ефективною і може допомогти менеджеру уникнути людських помилок та покращити розвиток бізнесу, у цій системі транзакція замовлення є поетапною моделлю, щоб зробити транзакцію більш систематизованою, і система може направляти персонал уникнути помилок у замовленні. Окрім ефективного обслуговування, за допомогою цієї системи вона може забезпечити якісніші послуги для клієнтів і залучить більше клієнтів до ресторану для отримання такої якості послуг. Ця система використовує мобільний додаток для прийому замовлень. Замовлення клієнта надсилатимуться на кухню через мобільний пристрій. Офіціантам не потрібно йти на кухню або в барний відділ, щоб дати замовлення, оскільки списки замовлень відображатимуться на екрані відділу. Операція між офіціантом та відділами ресторанів, а також між офіціантом та касиром буде систематичною.

Цією системою користуватиметься менеджер, персонал ресторану та клієнти. Керівник ресторану може контролювати роботу персоналу та ресторану в цілому. Персонал ділиться на двох. Один - офіціант, який приймає замовлення для клієнта, а інший - шеф-кухар, який працює в кухонному відділі. Шеф-кухар може бачити списки замовлень, сортуючи за номером таблиці, який

відображається на екрані. Це може допомогти зробити систему замовлення більш ефективною та уникнути людських помилок. Для адміністрації це може допомогти адміністратору керувати запасами по всій системі.

Є кілька різних способів навести порядок, який використовують ресторани. Зазвичай існує два способи наведення порядку, які використовують багато ресторанів, це ручна система та комп'ютеризована система. Деякі ресторани в Україні досі використовують ручну систему наведення порядку, а також є деякі ресторани, які використовують комп'ютеризовану систему замовлення. Ручна система подібна до того, що офіціант ресторану використовує ручку та папір для прийняття замовлення. Зазвичай ресторан робить дві копії списку замовлень. Один з примірників буде надісланий до кухонного відділу для складання кухонного списку, а інші примірники - у касу для розрахунку рахунків. Але в наш час у багатьох ресторанах використовується комп'ютеризована система, що стосується все більшої кількості нових технологій.

Ця комп'ютеризована система замовлення все ще використовує ручну систему для прийняття замовлення, наприклад, використовуючи ручку та папір для оформлення замовлення. Потім список замовлень буде введено в комп'ютер і збережено в базі даних комп'ютера. Після цього, список замовлення він відправить на кухонний принтер, щоб роздрукувати його. Список замовлення, який зберігався в комп'ютері, також буде надісланий до каси для розрахунку рахунків та роздрукування для клієнта. Ця комп'ютеризована система широко використовується у багатьох ресторанах, більшість із яких є ресторанами швидкого харчування. Ця комп'ютеризована система зробила систему замовлення більш систематизованою та швидшою.

На сьогоднішній день кожне підприємство прагне автоматизувати всі свої бізнес-процеси для успішного і ефективного ведення бізнесу. Автоматизація бізнес-процесів - це переклад типових бізнес-завдань і стандартних операцій під контроль програмно-апаратного комплексу. В результаті вивільняються ресурси, що дозволяє збільшити продуктивність праці і ефективність стратегічного управління. Для оптимізації роботи організації використовують системи



моделювання та оптимізації виконання бізнес-процесів. Тому створення і розвиток інформаційної інфраструктури є одним із пріоритетних завдань будь-якої організації. Необхідно розуміти, що тільки добре розвинене підприємство може успішно впоратися з сучасними темпами розвитку ринку і кількістю конкурентів, наявних на ньому. Впровадження інформаційних систем є відмінним інструментом для успішної організації діяльності будь-якої компанії. В Україні ресторанний бізнес з кожним днем розвивається все ширше і ширше. У зв'язку зі змінами в законодавстві, виникненням великого числа ресторанів і зростанням конкуренції потрібні нові методи, новий підхід до вирішення питань, пов'язаних з роботою ресторану.

Ресторанний бізнес одна з сучасних перспективно розвиваючих галузей індустрії гостинності. Люди приходять в ресторан не тільки скуштувати вишукані страви, а й відпочити, отримати естетичне задоволення і просто поспілкуватися один з одним. Інновації - один з основних двигунів ресторанного бізнесу. Якщо мислити глобально, то сьогодні Україна стоїть на порозі інформаційного суспільства. Відповідно, і інновації в секторі послуг повинні лежати в цій площині. Варіант для ресторанів - електронне меню і дистанційне керування послугами. Електронне меню ресторану - інтерактивна система замовлень, яка представляє собою сенсорний екран, планшет, наприклад, *iPad*, на якому можна отримати вичерпну інформацію по кожній страві на десятці мов: зовнішній вигляд, ціна, список інгредієнтів, калорійність, час приготування, докладний опис процесу приготування страви. Актуальність теми дослідження даної роботи полягає в тому, що ресторанам, які особливо стрімко розвиваються на ринку, досить проблематично контролювати прийом і обробку замовлень на належному рівні. Дана проблема вирішується шляхом впровадження електронного меню, яке допоможе підвищити ефективність ресторанного бізнесу і рівень сервісу. Метою випускної роботи є аналіз та удосконалення процесу обробки замовлень в ресторані.

Для виконання поставленої мети необхідне рішення наступних завдань:

- 1) вивчити ІТ-технології в ресторанному бізнесі;

- 2) вивчити види і особливості електронних меню, представлених на ринку;
- 3) порівняти функціонал розглянутих інформаційних систем;
- 4) вивчити структуру і діяльність ресторану;
- 5) проаналізувати існуючі бізнес-процеси на даному підприємстві, побудувати моделі *as-is, to-be*.

Онлайн-система замовлення, яку я пропоную тут, значно спрощує процес замовлення як для клієнта, так і для ресторану. Система представляє інтерактивне та сучасне меню з усіма доступними опціями у зручному для використання режимі. Клієнт може вибрати один або кілька предметів для розміщення замовлення, яке потрапить у кошик. Клієнт може переглянути всі деталі замовлення в кошику перед тим, як виписати товар. В кінці клієнт отримує деталі підтвердження замовлення. Після того, як замовлення розміщено, воно вноситься в базу даних і отримується майже в реальному часі. Це дозволяє працівникам ресторану швидко переглядати замовлення в міру їх отримання та ефективно обробляти всі замовлення з мінімальними затримками та плутаниною. Для побудови цього модуля використовуються мови: *JSP, HTML* та *Java* на стороні клієнта, тоді як база даних *MySQL* на задній панелі, оскільки я знайшов їх надзвичайно корисними під час роботи над технологіями.

**Мета дипломного проекту** – розробка та створення системи онлайн замовлень в ресторані.

**Об’єкт дослідження** – веб-система онлайн замовлень в ресторані.

Прогнозні припущення щодо розвитку об’єкта дослідження – створення системної утиліти та використання її в комп’ютерних системах, в яких є необхідність виявлення зміни параметрів файлів.

Результати дипломного проекту рекомендується використовувати в будь-якій автоматизованій системі ресторану.

## РОЗДІЛ 1

### АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

#### 1.1. Сучасні технології в задачах удосконалення процесів онлайн-замовлень з ресторанів

Ресторанний бізнес є такою сферою послуг, де необхідне застосування нових технологій. З огляду на прискорений темп сучасного життя, коли кожна секунда на рахунку, багато людей залишаються незадоволені якістю надаваних їм в ресторані послуг, а часом і не надаються, якщо в ресторані немає вільних місць. У боротьбі з цими проблемами можуть допомогти інформаційні технології. Одне з важливих напрямків розвитку ресторанних технологій - мобільні сервіси онлайн-бронювання столиків, які мають ряд переваг:

- мобільний телефон - завжди під рукою, і забронювати, наприклад, столик на вечір можна навіть по дорозі на роботу;

- всі дані про клієнтів і їх бронювання зберігаються не в книзі резервів, а в електронному вигляді, де все чітко структуроване;

- можливість вибрати на схемі залу стіл, який сподобався;

- немає комісії, на відміну від систем онлайн-бронювання на сайтах. Але, при цьому, є істотний мінус при бронюванні - клієнт «бронює» столик на конкретний час, а потім просто не приходить, при цьому ресторан зазнає збитків. Дану проблему вирішили Грант Акація і Нік Коконас, шеф-кухарі ресторанів *Alinea* і *Next* в Чикаго. Вони розробили систему бронювання - *Tock*, яка передбачає купівлю «квитків» в ресторан за аналогією з квитками в театр: користувач вибирає дату і час, ресторан, а потім оплачує все, що буде входити в замовлення.

Кафедра КСУ				НАУ 21 07 16 000 ПЗ			
Виконав	Чижик О. В.			Аналіз предметної області	Літера	Лист	Листів
Керівник	Нечипорук О.П.				Д	11	41
Консульт.					123 СП-435		
Н. контроль	Тупота Є.В.						
Зав. Каф.	Литвиненко О.Є.						

У підсумку, клієнт просто приходить в заклад і отримує те, за що вже заплатив.

Метою такого нововведення було збільшення швидкості обслуговування. Крім того, в деяких додатках є можливість поділу рахунку між декількома відвідувачами, що дозволяє клієнтам заплатити конкретну суму грошей або відсоток від загальної суми. Як це працює:

- гість авторизується в замовлення і натискає розділити рахунок;
- друзі гостя авторизуються в те ж замовлення;
- кожен гість вибирає страви, які буде оплачувати (сума до оплати змінюється для кожного гостя);

- кожен авторизований гість натискає до оплати і оплачує свій рахунок.

Сьогодні в Україні більше 30% операцій оплати рахунку в ресторанах і кафе відбувається за допомогою банківської карти. У найбільших містах ця частка перевищує 50%. Гості не носять з собою готівку. У зв'язку з цим дохід офіціантів почав стрімко знижуватися. Тому в даних мобільних додатках доступний функціонал «безготівкові чайові». Офіціанти зможуть отримувати винагороди від гостей на свої електронні гаманці в режимі онлайн.

Цінністю такої функції для ресторану є:

- збільшення доходу офіціантів, зменшення «плинності» персоналу;
- надання можливості для гостя залишити чайові навіть при відсутності готівки;
- збільшення частки гостей, які беруть активну участь в програмі лояльності.

Одне з останніх нововведень в ресторанному бізнесі - електронне меню. Електронне меню - інтерактивна система замовлень, що використовується в якості сучасної альтернативи звичному паперовому меню. Дана система містить повноцінну інформацію про страви та напої закладу на електронному пристрої. Крім того, відвідувачі ресторану самостійно можуть розглянути весь асортимент, представлений в ресторані і зробити своє замовлення, не чекаючи офіціанта. На відміну від звичного паперового меню електронне меню має ряд переваг:

- з екологічної точки зору, дана система дозволяє значно економити на папері, так як не потрібно заново передруковувати меню, кожен раз, коли з'являється необхідність у змінах;

- з точки зору маркетингу, електронне меню служить інструментом для залучення клієнтів, так як на сьогоднішній день інтерес людей до подібного роду технічних новинок помітно зростає;

- з точки зору обслуговування, істотно прискорюється процес прийому і обробки замовлення, виключаються помилки, що виникають із-за людського фактору;

- з точки зору автоматизації, електронне меню здатне знизити навантаження на персонал і організувати роботу кухарів, які виконують замовлення в порядку їх надходження. За типом установки електронні меню діляться на два види:

- стаціонарні меню;

Стаціонарні інтерактивні меню розташовані або на обідніх столах і прикріплені за допомогою спеціального тримача, або вбудовані безпосередньо в стіл. Таке меню допомагає заощадити час гостей на прихід офіціанта.

- виносні;

Таке меню відрізняється від друкованого тільки тим, що знаходиться на електронному пристрої - планшет.

В цілому, завдання будь-якого електронного меню зводиться до реалізації наступних бізнес-процесів:

- прийом замовлень;

- обробка замовлень;

- надання інформації про замовлення.

Серед великої різноманітності електронних меню найбільш популярними і розвиненими є: *eMenu*, *SmartMenu*, *ProfitMenu*. Статистика показує, що в тель-авівському суші-ресторані *Frame* дохід від столиків, оснащених *e-Menu*, виріс на 11% в порівнянні зі звичайними. У Києві, де вже є перші приклади впровадження подібних систем, конверсія (віддача від одного столика за зміну) столів

підвищилася в деяких ресторанах на 15%. Крім того, ресторатори зможуть економити на кадрах. Сьогодні в Харкові в середньому офіціант заробляє близько 100 *тис. грн.* на рік. Відповідно п'ять офіціантів - це 500 *тис. грн.* витрат на зарплату. Після впровадження електронного меню середній ресторан зможе якраз відмовитися від такої кількості людей.

## 1.2. Оптимізація бізнес-процесів на основі впровадження інформаційного додатку

Багато компаній стикаються з усвідомленням того, що зміни життєво необхідні для того, щоб залишатися на плаву і розвиватися, але далеко не всі з них представляють, у чому саме повинні полягати ці зміни. Яку ланку в ланцюжку варто замінити або усунути, щоб отримати максимальну віддачу. Причому, в більшості випадків, зміни передбачають не просто збільшення ефективності існуючих процесів, а трансформацію їх структури. Тільки підприємство, яке має чітке уявлення про те, який продукт або результат і в якому конкретно вигляді буде цікавий споживачеві, зможе знайти оптимальний шлях досягнення мети. Це вимагає сміливості, креативного мислення і, що важливо, готовності інвестувати в проект, так як зміна схеми управління, як правило, вимагає значних вкладень.

Підприємці прагнуть замінити дорогу людську працю на автоматичний, механізований, електронний підхід. При цьому автоматизація процесів не передбачає просте заміщення людей механізмами, це поступове визначення рутинних, повторюваних операцій в будь-якій професії і передача їх у відання електроніки. Що стосується сфери обслуговування, ресторан - заклад, що відрізняється від інших підвищеним рівнем обслуговування і приготуванням складних страв. Як і на будь-якому іншому підприємстві, які активно розвиваються, в ресторані дуже важливо налагодити роботу всіх бізнес-процесів, особливо прийом і обробку замовлень. Впровадження електронного меню допоможе повністю автоматизувати всі замовлення на належному рівні,

уникаючи помилок офіціанта і додаткових витрат, пов'язаних з виправленням цих недоліків, що призведе до:

- гарантії точності обслуговування;
- підняття іміджу ресторану;
- підвищення конкурентоспроможності;
- зниження навантаження на персонал.

Електронне меню може бути реалізовано одночасно на:

- планшетах, зафіксованих на кожному столі і, в кінці робочого дня, персонал зможе зняти їх і віднести на зарядку;
- виносних планшетах, які стануть в нагоді в разі, коли в якості відвідувачів прийде велика компанія;
- мобільних пристроях клієнтів, а саме в результаті підключення до певної точки *Wi-Fi*, автоматично відкривається сайт з меню.

Такий варіант використання системи допоможе охопити всіх гостей, і ніхто не залишиться без електронного меню. Надалі, підвищиться рівень сервісу в ресторані і лояльність клієнтів. ІС має на увазі повну автоматизацію прийому і обробки замовлень, а отже між гостем і кухарем ресторану не буде ніяких посередників в особі офіціанта або адміністратора. Замовлення безпосередньо через бездротову мережу *WiFi* будуть відправлятися на сенсорний екран, встановлений в кухні. Після закінчення приготування страви, кухар відзначить на екрані, що замовлення готове.

При цьому не потрібно завантажувати додаткове додаток для оплати замовлення безготівковим розрахунком, так як в даній системі можна розплатитися картою онлайн прямо в планшеті. Всі дані зберігаються в хмарному сховищі. Це забезпечує безпеку всієї інформації і транзакцій, що відбуваються в системі. У адміністратора є окремий доступ до ІС для того, щоб він міг редагувати і оновлювати меню, поміщати страви в стоп-лист, якщо по какимлибo причин вони не можуть бути приготовлені, а замість них відвідувачеві ресторану пропонуються їх аналоги. Також адміністратор зможе

контролювати роботу кожного офіціанта, отримувати статистичні дані та формувати звіти, що істотно підвищить ефективність ресторанного бізнесу.

### 1.3. Опис існуючих систем онлайн замовлень

*E-menu*. Протягом 5 років компанія *EMENU* займається оснащенням ресторанів системами автоматизації нового покоління. Одним з завдань досягнень є електронне меню. Існує 6 варіантів використання даної системи:

- *iMenu* - просте інтерактивне меню на базі сенсорного планшетного комп'ютера зі списком страв і кнопкою виклику офіціанта;
- *eMenu* на базі планшетів (електронне меню + система електронних замовлень);
- *eMenu* на базі сенсорного моноблока, який розміщується в залі поруч зі столиком (електронне меню + система електронних замовлень);
- *eMenu Table* - прозорий столик, в який вбудований сенсорний дисплей (електронне меню + система електронних замовлень);
- *eMenu Stand* - термінальна стійка (електронне меню + засіб реклами);
- *eMenu Glass* - сенсорний дисплей у вітрині закладу (електронне меню + засіб реклами).

Останні два варіанти використовується в кафе швидкого харчування, закусочних, фаст-фудах, тому їх розглядати не має сенсу. Так як метою є автоматизація замовлень, а саме пряму взаємодію користувача з кухарем без участі офіціанта, то із запропонованих варіантів підходить: *eMenu* на базі планшетів, *eMenu* на базі сенсорного моноблока і *eMenu Table*. Так само важливо враховувати, що впроваджувати цю систему будуть в уже існуючі ресторани, у яких немає необхідності змінювати концепцію свого закладу, витратити багато коштів для розробки проекту, а є завдання - поліпшити, прискорити і автоматизувати прийом і обробку замовлень. Отже, розглядати необхідно тільки *eMenu* на базі планшетів.



Для реалізації проекту по впровадженню електронного меню необхідні наступні додаткові компоненти:

- планшети;
- сервер (персональний комп'ютер);
- бездротова мережа *Wi-Fi*;
- принтер *eMenu*.

Для повноцінного функціонування ІС необхідна установка персонального комп'ютера, який відповідає за роль сервера, на якому повинна бути встановлена база даних і серверне ПЗ. Так як серверне програмне забезпечення приймає замовлення з планшетів і взаємодіє з уже існуючою в ресторані *POS*-системою, то ПК повинен бути завжди включений. Щоб уникнути перевантаження мережі, необхідно створити додаткову захищену бездротову мережу *Wi-Fi*, яка не буде використовуватися відвідувачами з метою виходу в Інтернет. Кількість точок доступу обов'язково повинні відповідати кількості використаних планшетів. У ролі контролера подій в системі є принтер чеків, підключений по мережі до сервера, на якому в процесі роботи реєструється такі події, як відправка замовлення, виклик офіціанта і запит рахунку.

Під час отримання запиту з планшета принтер друкує чек, який містить номер столика, дату та інші необхідні параметри. *Emenu* на базі планшетів включає в себе саме електронне меню і систему електронних замовлень і, завдяки цьому, його функціонал дуже широкий і різноманітний:

- можливість зробити замовлення без участі офіціанта, оскільки дані відразу відправляються на кухню;
- повна інформація про страви та напої, що включає в себе короткий опис, склад, калорійність і час приготування;
- безліч топінгів і спеціальних добавок, передбачених для кожного блюда;
- функція пошуку і сортування страв;
- проста система навігації;

- страви, які з яких-небудь причин не можуть бути приготовлені, поміщаються в стоп-лист, а замість них відвідувача ресторану пропонуються їх аналоги;

- кнопка виклику офіціанта;

- багатомовна підтримка;

- вбудована система крос-продажів, що рекомендує додаткові соуси і закуски до страви;

- для маленьких відвідувачів закладу є можливість вибрати і включити перегляд мультфільмів;

- доступ в Інтернет, соціальні мережі, будь-які ігри та програми під ОС *Android*;

- за допомогою *eMenu* є можливість просувати будь-яку продукцію і послуги компаній-партнерів, реклама яких з'являється після виключення користувачем пристрою.

Для реалізації проекту по впровадженню електронного меню необхідні наступні додаткові компоненти:

- планшети;

- сервер (персональний комп'ютер):

- бездротова мережа *Wi-Fi*;

- принтер *eMenu*.

Для повноцінного функціонування ІС необхідна установка персонального комп'ютера, який відповідає за роль сервера, на якому повинна бути встановлена база даних і серверне ПЗ. Так як серверне програмне забезпечення приймає замовлення з планшетів і взаємодіє з уже існуючою в ресторані *POS*-системою, то ПК повинен бути завжди включений. Щоб уникнути перевантаження мережі, необхідно створити додаткову захищену бездротову мережу *Wi-Fi*, яка не буде використовуватися відвідувачами з метою виходу в Інтернет. Кількість точок доступу обов'язково повинні відповідати кількості використаних планшетів. У ролі контролера подій в системі є принтер чеків, підключений по мережі до сервера, на якому в процесі роботи реєструється такі події, як відправка

замовлення, виклик офіціанта і запит рахунку. Під час отримання запиту з планшета принтер друкує чек, який містить номер столика, дату та інші необхідні параметри.

*Menu* на базі планшетів включає в себе саме електронне меню і систему електронних замовлень і, завдяки цьому, його функціонал дуже широкий і різноманітний:

- вбудована система крос-продажів, що рекомендує додаткові соуси і закуски до страви;

Система електронних замовлень *Profit Menu*, призначена для всіх мобільних пристроїв, розроблена в 2013 році. З самого початку продукт базувався в локальній інфраструктурі, але з часом компанія вирішила перейти на хмарну основу, що забезпечує 100% безпеку і приватність всіх транзакцій системи. Схема роботи електронного меню полягає в наступному: клієнт, що прийшов в ресторан, підключається зі свого мобільного пристрою до *Wi-Fi* і відразу потрапляє на сайт *tvoe.menu*, де йому надається повний список страв і напоїв. Після формування замовлення, ця інформація безпосередньо відправляється на кухню, а саме на екран, де в порядку черги розміщуються отримані замовлення. Також, рестораторові не потрібно витрачатися на підключення додаткової бездротової мережі *Wi-Fi*, так як це вже входить у вартість ліцензії.

Основною ідеєю *Profit Menu* є те, що гості закладу зможуть завжди скористатися персональним мобільним додатком миттєво доступному на їх мобільному пристрої без будь-яких установок.

Функціонал є наступним:

- докладна інформація про страви та напої, що включає в себе короткий опис, склад, калорійність і час приготування;
- сортування страв за категоріями;
- кнопка виклику офіціанта;
- є функція «перегляд статусу замовлення»;
- перед замовленням гостю необхідно ввести номер свого столика;

- можливо зайти в меню як в гостьовому режимі, так і як зареєстрований користувач;

- доступна функція «попросити рахунок у офіціанта»;

- багатомовна підтримка.

#### 1.4. Висновки до розділу

В даний час електронне меню для підприємств харчування практично повністю замінює традиційне і пропонує цілий ряд додаткових функцій. Воно не тільки є каталогом продажів і джерелом гастрономічних пропозицій, але і допомагає рекламувати і продавати запропонований товар, підвищує швидкість і усуває людський фактор при прийомі замовлення, вміє привертати увагу гостей, здатне переконувати, розважати і приносити задоволення. Електронне меню - це відмінний інструмент по залученню клієнтів і збільшення прибутку в руках ресторану, і незабутні враження в руках гостя. Підводячи підсумки, слід зазначити, що в кожній системі присутні як переваги, так і недоліки. Кожен розробник електронного меню має свої конкретні цілі, які він хоче досягти, створюючи даний проект. У будь-якому випадку, застосування призводить до збільшення швидкості обслуговування та конверсії столів, що свою чергу підвищує прибуток підприємства.

В рамках написання 1 розділу дипломної роботи:

- розглянуто *IT*-технології в ресторанному бізнесі;

- описані підходи до оптимізації бізнес-процесів на основі впровадження інформаційного додатку;

- вивчені види і особливості електронних меню, представлених на ринку;

- порівняні функціонали розглянутих інформаційних систем.

## РОЗДІЛ 2

### АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 2.1 Функціональні вимоги

Починаючи новий проект для того, щоб він згодом перетворився на величезну систему, ми знаємо, що чіткі визначені вимоги мають вирішальне значення для успіху. Визначення вимог заздалегідь дозволяє ефективно управляти очікуваннями клієнтів і створювати плавний та безперебійний процес впровадження.

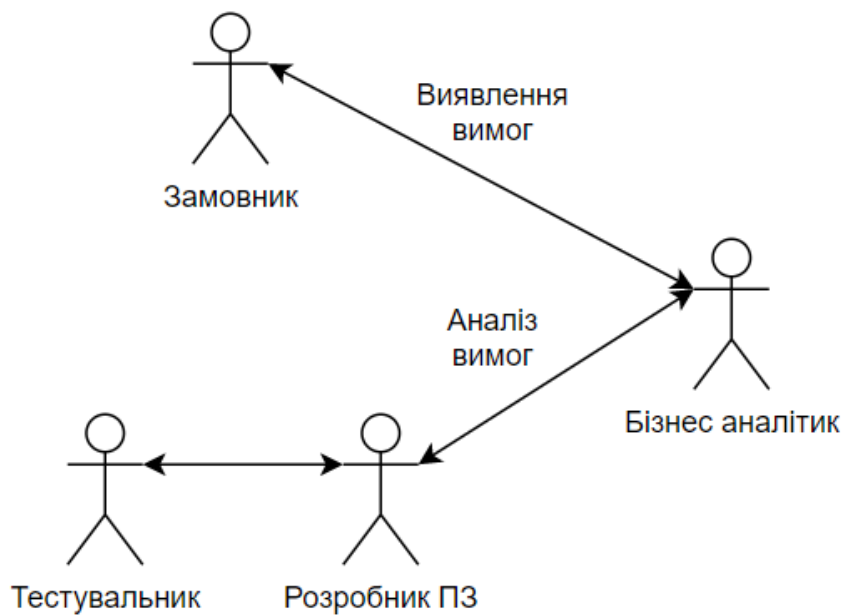


Рис. 2.1. Схема виявлення та аналізу вимог до ПЗ.

Кафедра КСУ				НАУ 21 07 16 000 ПЗ			
Виконав	Чижик О.В.			Аналіз вимог до програмного забезпечення	Літера	Лист	Листів
Керівник	Нечипорук О.П.				Д	21	41
Консульт.					123 СП-435		
Н. контроль	Тупота Є.В.						
Зав. Каф.	Литвиненко О.Є.						

Однак проект, якому не вистачає цієї чіткості, може призвести до неясно визначеного обсягу проекту та результатів, які не відповідають оцінці. Від 60 до 80 відсотків вартості розробки програмного забезпечення переробляється. Крім того, успішне управління вимогами може усунути від 50 до 80 відсотків дефектів проекту.

Бізнес-аналітики відповідають за подолання розриву між ІТ та бізнесом за допомогою аналізу даних для оцінки процесів, визначення вимог та надання рекомендацій та звітів на основі даних керівникам та зацікавленим сторонам.

Бізнес-аналітики взаємодіють з керівниками підприємств та користувачами, щоб зрозуміти, як зміни на основі даних, що обробляються, продукти, послуги, програмне та апаратне забезпечення можуть підвищити ефективність та надати додаткову цінність. Вони повинні сформулювати ці ідеї, але також збалансувати їх з технологічно доцільним, фінансово та функціонально обґрунтованим. Залежно від ролі, ви можете працювати з наборами даних для вдосконалення продуктів, обладнання, інструментів, програмного забезпечення, послуг чи процесів.

Роль бізнес-аналітика постійно розвивається та змінюється, оскільки компанії більше покладаються на дані, щоб консультувати бізнес-операції. Кожна компанія має різні проблеми, які може вирішити бізнес-аналітик, незалежно від того, має справу з застарілими системами, зміною технологій, непрацездатністю процесів, поганим задоволенням клієнтів чи споживачів чи закритими великими організаціями.

Проаналізувавши вимоги замовника, складений наступний перелік функціональних вимог:

1. Система повинна мати сторінку реєстрації користувачів.
2. На сторінці реєстрації користувач має бачити поля для вводу логіну, пароля та *email*.
3. На сторінці реєстрації користувач має бачити кнопку зареєструватися за допомогою *Google*.

4. При введенні некоректних даних для реєстрації, користувач має бачити повідомлення, що пішло не так.
5. Після успішної реєстрації система повинна перенаправити користувача на сторінку входу.
6. На сторінці входу повинно бути два поля для введення логіну та паролю.
7. На сторінці входу повинна бути кнопка – “Забули пароль?”
8. На сторінці “Забули пароль?” користувач повинен ввести свою *email* адресу.
9. На сторінці “Забули пароль?”, після введення *email*, система повинна перевірити існуючого користувача.
10. Після перевірки існуючого користувача, система повинна надіслати на пошту користувача посилання на відновлення паролю.
11. Після успішного входу система повинна повернути головну сторінку.
12. На головній сторінці користувач повинен бачити пункт “Меню”.
13. На головній сторінці користувач повинен бачити пункт “Вихід”.
14. На головній сторінці користувач повинен бачити пункт “Зробити замовлення”.
15. На головній сторінці користувач повинен бачити пункт “Мої замовлення”.
16. На сторінці “Меню” користувач повинен бачити доступне меню для того щоб зробити замовлення.
17. На сторінці “Меню” користувач повинен мати змогу перейти на вкладку “Зробити замовлення” .
18. На сторінці “ Зробити замовлення ” користувач повинен мати змогу зробити замовлення з вибору доступних позицій.
19. На сторінці “ Зробити замовлення ” користувач має бачити кнопку підтвердити замовлення.
20. Після підтвердження замовлення сторінка “Мої замовлення” має містити список всіх замовлень користувача.

21. При натисканні кнопки Вихід, система повинна закрити активну сесію користувача.

## 2.2. Нефункціональні вимоги

Нефункціональні вимоги мають вирішальне значення для зручності використання програмної системи, і якщо ви не будете їх ретельно визначати, це може негативно вплинути на досвід кінцевих користувачів. Прикладом нефункціональних вимог є визначення того, як швидко веб-сайт повинен завантажуватися, або зазначення, що веб-сайт повинен обробляти 10 мільйонів користувачів, не маючи жодних проблем із продуктивністю. Нефункціональні вимоги також можуть бути пов'язані з безпекою; наприклад, користувач повинен змінити початковий пароль для входу під час першого входу. Інші приклади включають такі, що стосуються:

- відповідність;
- документація;
- конфіденційність;
- переносимість;
- якість;
- надійність;
- стійкість;
- час реакції;
- масштабованість;
- стабільність;

Нефункціональні вимоги зосереджені на роботі системи, а не на її поведінці; наприклад, дані повинні оновлюватися для користувачів під час доступу протягом трьох секунд. Нефункціональні вимоги також можуть бути визначені метриками та стосуватися аспектів системи, що еволюціонують з часом, таких як керування чи документація.



Порівнюючи функціональні та нефункціональні вимоги, слід враховувати, що функціональні вимоги можуть забезпечити завантаження системи певної веб-сторінки, коли користувач натискає кнопку. Нефункціональна вимога може визначати швидкість завантаження сайту. Веб-сайт, який завантажується повільно, може негативно вплинути на взаємодію з користувачем, саме тому нефункціональні вимоги є критичними.

Проаналізувавши вимоги замовника, складений наступний перелік нефункціональних вимог:

- принаймні 20 відсотків ємності процесора та місця для зберігання, доступних системі, не повинні використовуватися при піковому навантаженні;
- цикл перезапуску системи повинен повністю виконатись менш ніж за 60 секунд;
- система повинна мати можливість обробляти повідомлення за 1 секунду або менше, і до 100 повідомлень включно за 15 секунд або менше;
- початкова система повинна мати можливість обробляти замовлення клієнтами з мінімальною швидкістю 10 в секунду;
- система повинна приймати 300 одночасних користувачів або менше в період пікового навантаження з 9:00 до 11:00 ранку;
- максимальне одночасне завантаження ємності користувача в непікові періоди становитиме 150;
- будь-який інтерфейс між користувачем та автоматизованою системою повинен мати максимальний час відгуку дві секунди;
- повні короткі звіти про замовлення з ресторану поточного робочого дня будуть доступні через хвилину після закінчення дня;
- постійне технічне обслуговування, яке виконується під час активності користувачів, не повинно спричиняти помітного збільшення часу відгуку для будь-якої функції, яка перевищує 5% за час відгуку, коли жоден процес обслуговування не виконується;
- жодна частина тексту, яка може відобразитися користувачеві, не повинна міститися у вихідному коді програми. Кожен фрагмент тексту, який користувач

може побачити, має змінюватися без зміни вихідного коду. Тобто жоден видимий для користувача текст не буде «*hard coded*». У діаграмах потоків даних всі використовувані символи складаються в загальну картину, яка дає чітке уявлення про те, які дані використовуються, і які функції виконуються.

Потоки даних формуються наступним чином: клієнт подає заявку на підключення до мережі Інтернет, її обробляють співробітники фірми: при цьому інформація про клієнта додається в базу даних, проводиться запит на сервер для виділення фізичної *IP* адреси, інформація про клієнта також заноситься в базу даних телефонної компанії. Далі надходить заявка майстрам на монтаж устаткування, подальше тестування мережі і в підсумку запускається автоматизований процес тарифікації, після чого клієнт вносить авансовий платіж, формуючи баланс на наступний місяць.

На кожній вищевикладеній стадії накопичується різна інформація, необхідна для грамотного менеджменту. Ця інформація відправляється в сховище даних, звідки по нерегламентованого запиту надходить до начальства. Різні звіти також відправляються до начальства з метою контролю роботи.

### 2.3. Висновки до розділу

Якість нефункціональних вимог безпосередньо визначає якість продукту, що розробляється або системи і досягається за рахунок ітеративного процесу визначення і аналізу нефункціональних вимог при злагодженій роботі всієї групи, яка бере участь в їх розробці.

## РОЗДІЛ 3

### СТРУКТУРА СИСТЕМИ ОНЛАЙН ЗАМОВЛЕНЬ В РЕСТОРАНІ

#### 3.1. Серверна частина системи

Майбутня система буде побудована за сервлетною технологією та розгорнутою на контейнері сервлетів *Tomcat*.

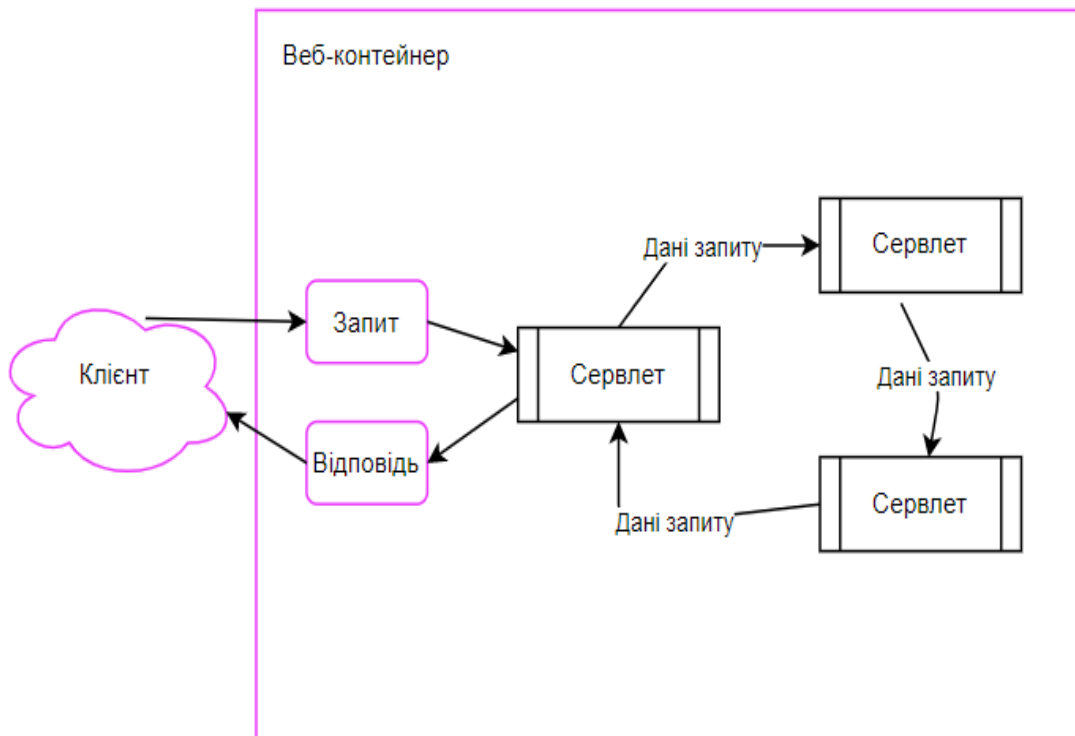


Рис. 3.1. Розгортання серверу *Tomcat*

Кафедра КСУ				НАУ 21 30 85 000 ПЗ			
Виконав	Чижик О.В.			Структура системи онлайн замовлень в ресторані	Літера	Лист	Листів
Керівник	Нечипорук О.П.				Д	27	41
Консульт.					123 СП-435		
Н. контроль	Тупота Є.В.						
Зав. Каф.	Литвиненко О.С.						

*Apache Tomcat*. *Apache Tomcat* - це найпопулярніший фактичний сервер веб-додатків *Java* із відкритим кодом, стандартний для сучасних веб-розробників, що використовують *JSP* / сервлети. *Apache Tomcat 7* охоплює подробиці щодо встановлення та адміністрування *Apache Tomcat 7*. У ньому роз'яснюються ключові частини архітектури *Tomcat* та пропонується вступ до *API Java Servlet* та *JSP* в контексті сервера *Apache Tomcat*.

На додаток до основних концепцій та завдань адміністрування, *Apache Tomcat 7* охоплює деякі найбільш часто використовувані розширені функції *Tomcat*, включаючи безпеку, інтеграцію веб-сервера *Apache*, балансування навантаження та вбудовування сервера *Tomcat* у додатки *Java*.

Сторінка *Java Server (JSP)* - це технологія управління вмістом або зовнішнім виглядом веб-сторінок за допомогою сервлетів, невеликих програм, які вказані на веб-сторінці та запускаються на веб-сервері для модифікації веб-сторінки перед її надсиланням на користувача, який його запитав. *Sun Microsystems*, розробник *Java*, також називає технологію *JSP* інтерфейсом прикладної програми *Servlet (API)*. *JSP* можна порівняти з технологією *Active Server Page (ASP)* від *Microsoft*. У той час як сторінка *Java Server* викликає програму *Java*, яка виконується веб-сервером, сторінка *Active Server* містить сценарій, який інтерпретується інтерпретатором сценаріїв (наприклад, *VBScript* або *JScript*) перед тим, як сторінка буде надіслана користувачеві.

Тобто все, що можна зробити за допомогою *JSP*, також може бути здійснено за допомогою сервлетів *Java*. Однак важливо зазначити, що сервлети та *JSP* є додатковими технологіями, а не заміною один одного. Сервлет можна розглядати як "*HTML* всередині *Java*", що краще для реалізації бізнес-логіки - оскільки він є домінуючим у *Java*. *JSP*, з іншого боку, є "*Java* всередині *HTML*", що є кращим для створення презентацій - оскільки це *HTML*-домінанта. У типовій програмі *Model-View-Control (MVC)* сервлети часто використовуються для контролера (C), що передбачає складну логіку програмування. *JSP* часто використовуються для подання (V), яке в основному стосується презентації. Модель (M) зазвичай реалізується за допомогою *JavaBean* або *EJB*.

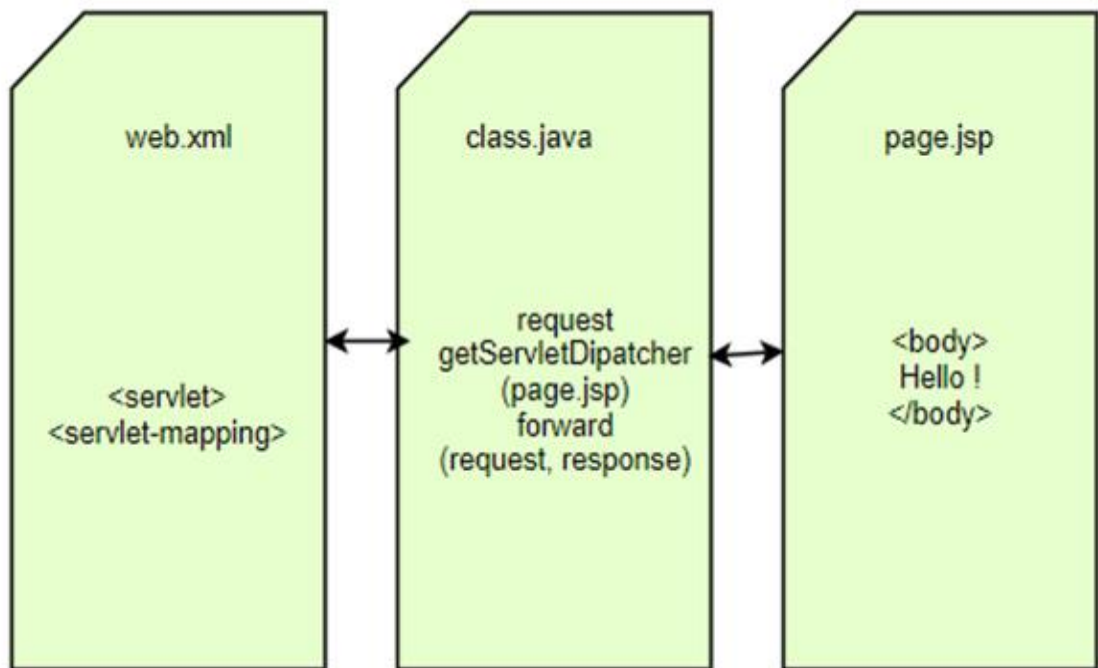


Рис. 3.2. Розгортання серверу *Tomcat*

На рисунку 3.2 зображена структура взаємодія *java*-класів та *JSP*. У файлі *web.xml* вказаний мапінг сервлетів, тобто на який *URL* вони будуть реагувати. В самому класі *java* приймається *http* запит, та відправляється далі на відображення до сторінки *JSP*.

*JSTL*. Спочатку веб-дизайнери використовували скриплетів на сторінках *JSP* для створення динамічного контенту. Це призвело до проблем з читабельністю, а також ускладнило ведення сторінки *JSP*. Власні теги були введені для подолання проблем, що виникають при використанні скриплетів. Хоча користувацькі теги виявилися кращим вибором, ніж скриплетів, вони також мали певні обмеження. Веб-дизайнерам довелося витратити багато часу на кодування упаковки та тестування цих тегів, перш ніж використовувати їх. Це означало, що веб-дизайнерам часто залишалося мало часу, щоб сконцентруватися на розробці веб-сторінок.

Впровадження *JSTL* допомогло веб-дизайнерам подолати недоліки користувацьких тегів, інкапсулюючи загальні функції, які веб-дизайнеру можуть знадобитися для розробки веб-сторінок. Ці функції включали використання

бібліотек тегів, таких як ядро, *SQL* та *XML*. *JSTL* представлений спеціально для тих веб-дизайнерів, які недостатньо добре обізнані з програмуванням на *Java*. *JSTL* 1.2, представлений на платформі *Java EE 5*, відповідає уніфікованій *EL*. Зверніть увагу, що уніфікований *EL* допомагає *JavaServer Faces (JSF)* використовувати теги *JSTL*. Така ж версія *JSTL*, тобто *JSTL* 1.2, використовується і на платформі *Java EE 6*.

### 3.2. База даних

За одним із пунктів нефункціональних вимог, майбутня система має бути з'єднаною з базою даних *MySQL*.

*MySQL* - це реляційна система управління базами даних (СУБД), заснована на запитах *SQL* (мова структурованих запитів). Це одна з найпопулярніших мов для доступу та керування записами в таблиці. *MySQL* є відкритим та безкоштовним програмним забезпеченням під ліцензією *GNU*. Компанія *Oracle* підтримує її.

*MySQL* - це реляційна система управління базами даних. Ця мова бази даних базується на запитах *SQL* для доступу та керування записами таблиці.

Проста у використанні. *MySQL* проста у використанні. Ми повинні отримати лише базові знання *SQL*. Ми можемо створювати та взаємодіяти з *MySQL*, використовуючи лише кілька простих операторів *SQL*.

Це безпечно. *MySQL* складається з надійного рівня захисту даних, який захищає конфіденційні дані від зловмисників. Крім того, паролі шифруються в *MySQL*.

Клієнт-серверна архітектура. *MySQL* слідує за роботою архітектури клієнт - сервер. Існує сервер баз даних *MySQL* і доволі багато клієнтів (прикладних програм), які взаємодіють із сервером; тобто вони можуть запитувати дані, зберігати зміни тощо.

Безкоштовно завантаження. *MySQL* можна використовувати безкоштовно, щоб ми могли завантажити його з офіційного веб-сайту *MySQL* без будь-яких витрат.

Це масштабовано. *MySQL* підтримує багатопоточність, що робить його легко масштабованим. Він може обробляти майже будь-який обсяг даних, до 50 мільйонів рядків і більше. Розмір файлу за замовчуванням становить близько 4 ГБ. Однак ми можемо збільшити це число до теоретичної межі 8 ТБ даних.

Швидкість. *MySQL* вважається однією з найшвидших мов баз даних, підкріпленою великою кількістю тестових тестів.

Висока гнучкість. *MySQL* підтримує велику кількість вбудованих програм, що робить *MySQL* дуже гнучким.

Структура таблиць системи онлайн замовлень з ресторану виглядає наступним чином:

Таблиця 3.1

Опис бази даних

Назва таблиці в БД	Опис
<i>custom</i>	Містить замовлення від користувачів
<i>status</i>	Зберігає статуси замовлення
<i>users</i>	Містить всіх користувачів системи
<i>roles</i>	Містить ролі користувачів
<i>food</i>	Містить список всіх страв ресторану
<i>employees</i>	Містить список працівників ресторану
<i>feedbacks</i>	Містить список відгуків

Для створення таблиць використано скрипти ініціалізації при запуску сервлетного веб-додатку.

```
CREATE TABLE EMPLOYEES
```

```
(
```

```
ID BIGINT NOT NULL,  
FIRST_NAME VARCHAR(30) NOT NULL,  
LAST_NAME VARCHAR(30) NOT NULL,  
DATE_OF_BIRTH DATE NOT NULL,  
SALARY INT NOT NULL
```

)

Дана таблиця зберігає інформацію про працівників ресторану, їх персональні дані та рівень заробітної плати.

```
CREATE TABLE CUSTOM
```

```
(
```

```
    ID BIGINT NOT NULL,  
  
    USER_ID BIGINT NOT NULL,  
    PRICE VARCHAR(30) NOT NULL,  
    TYPE VARCHAR NOT NULL  
    USER_ID FOREIGN KEY REFERENCES USERS(id)
```

```
)
```

Дана таблиця зберігає інформацію про замовлення користувачів, а саме в ній записується ціна, ім'я користувача, який зробив замовлення та тип замовлення.

```
CREATE TABLE FEEDBACKS
```

```
(
```

```
    ID BIGINT NOT NULL,  
  
    USER_ID BIGINT NOT NULL,  
    TEXT VARCHAR(30) NOT NULL,  
    USER_ID FOREIGN KEY REFERENCES USERS(id)
```

```
)
```



### 3.3. Діаграма класів

На рисунку 3.3 зображена діаграма класів системи онлайн-замовлень з ресторану.

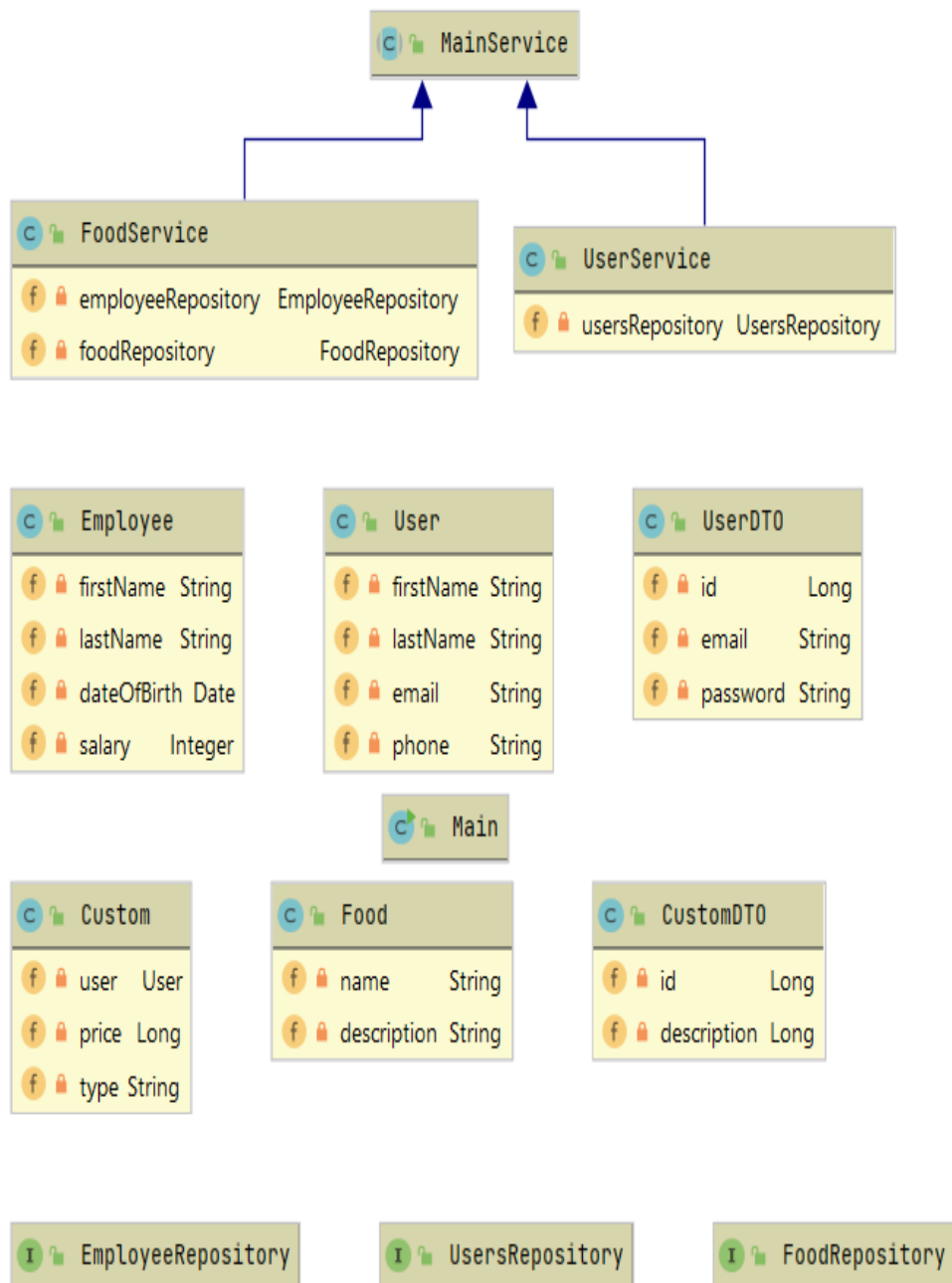


Рис. 3.3. Діаграма класів

Структура пакетів даного проекту побудована за шаблоном *MVC*, також присутній пакет *repository*, в якому зберігаються інтерфейси для доступу до бази даних (рис. 3.4).

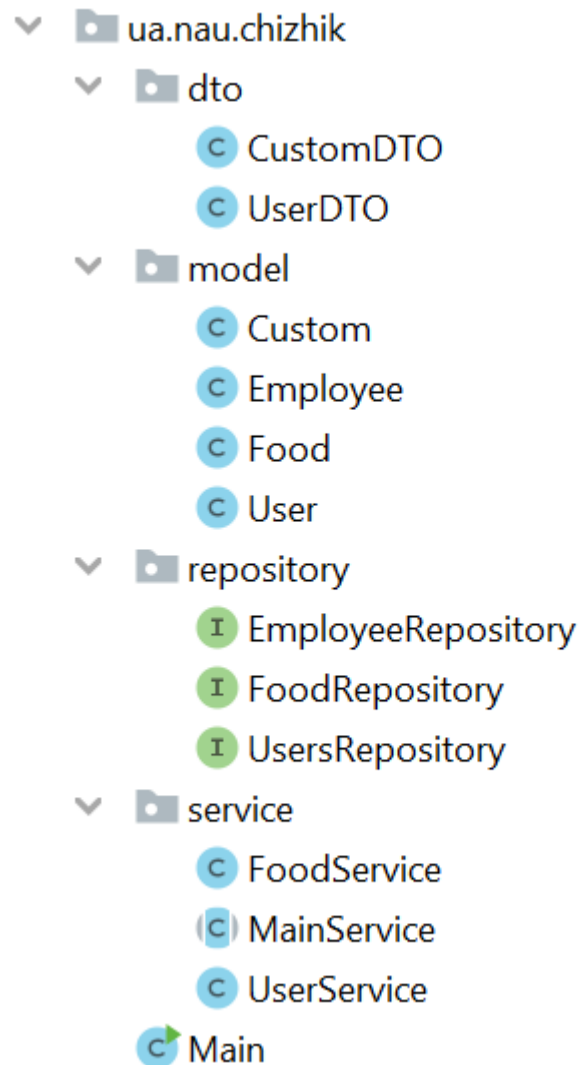


Рис. 3.5. Структура пакетів проекту

Пакети *Java* - це механізм групування класів *Java*, які пов'язані між собою, в одну і ту ж "групу" (пакет). Коли проект *Java* зростає, наприклад, програма або *API*, корисно розділити код на кілька класів *Java*, а класи на кілька пакетів *Java*.

Коли розділяються класи на кілька пакетів *Java*, стає легше зрозуміти, де знаходиться певний клас, який потрібно знайти (табл. 3.2).

Таблиця 3.2

Опис класів проекту

Файл <i>Java</i>	Опис
<i>CustomDTO.java</i>	Модель даних замовлення при <i>http</i> обміну з <i>JSP</i> сторінками.
<i>UserDTO.java</i>	Модель даних користувачів при <i>http</i> обміну з <i>JSP</i> сторінками.
<i>Custom.java</i>	Модель “Замовлення”
<i>Employee.java</i>	Модель “Працівник”
<i>Food.java</i>	Модель “Страва”
<i>User.java</i>	Модель “Користувач”
<i>EmployeeRepository.java</i>	Інтерфейс для обміну даними з таблицею “ <i>employee</i> ”, яка знаходиться на віддаленому сервері <i>MySQL</i> .
<i>FoodRepository.java</i>	Інтерфейс для обміну даними з таблицею “ <i>food</i> ”, яка знаходиться на віддаленому сервері <i>MySQL</i> .
<i>UsersRepository.java</i>	Інтерфейс для обміну даними з таблицею “ <i>users</i> ”, яка знаходиться на віддаленому сервері <i>MySQL</i> .
<i>FoodService.java</i>	Сервіс для реалізації бізнес логіки, зв’язаної з замовленням страв.
<i>UserService.java</i>	Сервіс для реалізації бізнес логіки, зв’язаної з діями користувачів.
<i>MainService.java</i>	Абстрактний батьківський клас для винесення загальної логіки всіх сервісів.

### 3.4. Висновки до розділу

На додаток до основних концепцій та завдань адміністрування, *Apache Tomcat 7* охоплює деякі найбільш часто використовувані розширені функції *Tomcat*, включаючи безпеку, інтеграцію веб-сервера *Apache*, балансування навантаження та вбудовування сервера *Tomcat* у додатки *Java*.

Отже, система онлайн замовлень в ресторані побудована та розгорнута на сервері *Tomcat*. В свою чергу в ньому розгорнуті сервлетні сторінки за допомогою технології *JSP*.

## ВИСНОВКИ

У дипломному проекті на тему «Програмний модуль системи онлайн замовлень в ресторані» була реалізована система онлайн замовлень в ресторані для того щоб клієнти могли робити замовлення на будь яку страву ресторану. Для цього було проаналізовано можливості інформаційних систем ресторанного бізнесу. Було проведено дослідження понять онлайн система, веб-додаток і їх місця у ресторанному бізнесі.

В даний час електронне меню для підприємств харчування практично повністю замінює традиційне і пропонує цілий ряд додаткових функцій. Воно не тільки є каталогом продажів і джерелом гастрономічних пропозицій, але і допомагає рекламувати і продавати запропонований товар, підвищує швидкість і усуває людський фактор при прийомі замовлення, вміє привертати увагу гостей, здатне переконувати, розважати і приносити задоволення. Електронне меню - це відмінний інструмент по залученню клієнтів і збільшення прибутку в руках ресторану, і незабутні враження в руках гостя. Підводячи підсумки, слід зазначити, що в кожній системі присутні як переваги, так і недоліки. Кожен розробник електронного меню має свої конкретні цілі, які він хоче досягти, створюючи даний проект. У будь-якому випадку, застосування призводить до збільшення швидкості обслуговування та конверсії столів, що свою чергу підвищує прибуток підприємства.

В рамках написання дипломної роботи:

- розглянуто *IT*-технології в ресторанному бізнесі;
- описані підходи до оптимізації бізнес-процесів на основі впровадження інформаційного додатку;
- вивчені види і особливості електронних меню, представлених на ринку;

- порівняні функціонали розглянутих інформаційних систем.

Після цього виявлено функціональні і нефункціональні вимоги до веб-системи.

Якість нефункціональних вимог безпосередньо визначає якість продукту, що розробляється або системи і досягається за рахунок ітеративного процесу визначення і аналізу нефункціональних вимог при злагодженій роботі всієї групи, яка бере участь в їх розробці.

Подальше використання розробленого продукту можливе у напрямку його використання в інформаційній системах, в яких є можливість зробити замовлення.

Результати дипломного проекту рекомендується використовувати при розробці програмних засобів, призначених для систем ресторанного бізнесу.

## СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ 3008-95 Структура і правила оформлення. Документація. Звіти у сфері науки і техніки
2. Бойченко С.В., Іванченко О.В. Положення про дипломні роботи (проекти) випускників Національного авіаційного університету. – К.: НАУ, 2017. – 63 с.
3. *Carroll, J., "Computer Security", 2nd Edition, Butterworth Publishers, Stoneham, MA, 1987.*
4. *Shirey, R., "Defense Data Network Security Architecture", Computer CommUNICATION Review, Vol. 20, No. 2, Page 66, 1 April 1990.*
5. Андрійчук А.В. Программирование в среде *Java*. Самоучитель, М.: *BHV* – 2014, 233 с.
6. Антонов Б.А., Калинов Ф.В. Програмное обеспечение. СПб.: Питер, 2001, 345 с..
7. Пічук Ф.А., Слісаренко А.К. Программирование на языке Джава: Учебное пособие. – Тула, 2015. – 169 с.
8. Манур О.О. Ресторанный бизнес онлайн. – СПб: ЗАО «Издательство «Питер», 2005. – 112 с.
9. Кутут П. «Быстрая разработка программного обеспечения», М: Лори, 2014. – 25.
10. Константайн Л. «Разработка программного обеспечения», Сп-б: «Питер», 2014. – 232 с.
11. Фогель С.К. Информационные системы, М.: *BHV*. – 2012. – 121 с.
12. Перес Франк. Проектирования баз данных.: Пер. с англ. – М.: Мир, 2016. – 124 с.
13. Купин В. «Технологии разработки программного обеспечения. Изд.3», Сп-б: «Питер», 2009., 124 с.

14. Гумен Штонер, Возможности реляционных баз данных 2012, – СПб.: Питер, 2012 – 221 с.
15. Криптол Ван «Инженерия программного обеспечения. Изд.6»; М: Диалектика-Вильямс, 2009, 123с.



## Додаток А

Лістинг коду модулю авторизації користувача в системі онлайн замовлень

```
@WebServlet("/signIn")
public class SignInServlet extends HttpServlet {

    private UserAuthService userAuthService;

    @Override
    public void init(ServletConfig config) throws ServletException {
        ServletContext context = config.getServletContext();
        userAuthService = (UserAuthService)context.getAttribute("usersService");
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        request.getRequestDispatcher("login.jsp").forward(request, response);
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        String email = request.getParameter("email");
        String password = request.getParameter("password");

        SignInRequest signInRequest = new SignInRequest(email, password);
        HttpSession session = request.getSession();
```

## Додаток Б

### ЛІСТИНГ КОДУ МОДЕЛІ КОРИСТУВАЧА

```
public class User {  
  
    private Integer id;  
    private String username;  
    private String email;  
    private String password;  
    private ERole role;  
  
    public User(String username, String email, String password) {  
        this.username = username;  
        this.email = email;  
        this.password = password;  
    }  
  
    public User(Integer id, , String password, ERole role) {  
        this.id = id;  
        this.username = username;  
        this.email = email;  
        this.password = password;  
        this.role = role;  
    }  
  
    public User(String username, String email, String password, ERole role) {  
        this.username = username;  
        this.email = email;  
        this.password = password;
```

```
        this.role = role;
    }

    public User(Integer id, String username, String email, String password) {
        this.id = id;
        this.username = username;
        this.email = email;
        this.password = password;
    }

    public User(String username, String email) {
        this.username = username;
        this.email = email;
    }

    public User(String username) {
        this.username = username;
    }

    public User() {
    }
}
```