

Міністерство освіти і науки України
Одеський національний політехнічний університет
Інститут комп'ютерних систем
Кафедра інформаційних систем

КУРСОВА РОБОТА

з дисципліни «Технології створення програмних продуктів»

за темою

«Інформаційна автоматизація роботи кафе»

Частина №2

Виконав:

студент 3 курсу

групи AI-181

Горбик М.М.

Перевірив:

Блажко О.А.

Одеса 2020

Анотація

В курсовій роботі розглядається процес створення програмного продукту «CWOS» (Cafe work organization service або Сервіс з онлайн організації роботи кафе). Робота виконувалась в команді з трьох учасників: Архіпов Д, Горбик М. та Козиревич Я. Тому в пояснювальній записці у розділах «Проектування» та «Конструювання» детальніше описано лише одну частину з урахуванням планів проведених робіт з розділу «Планування» з описом особливостей конструювання:

- структур даних реляційної моделі в системі керування базами даних PostgreSQL;
- програмних модулів в інструментальному середовищі IntelliJIDEA Ultimate з використанням фреймворку Spring та мови програмування Java. Результати роботи розміщено на github-репозиторії за адресою:

Перелік скорочень

ОС – операційна система

ІС – інформаційна система

БД – база даних

СКБД – система керування базами даних

ПЗ – програмне забезпечення

ПП– програмний продукт

UML – уніфікована мова моделювання

Клієнт – особа, що взаємодіє з ПП

ЗМІСТ

1.	Вимоги до програмного продукту	5
1.1	Визначення потреб споживача	5
1.1.1	Ієрархія потреб споживача	5
1.1.2	Деталізація матеріальної потреби.....	6
1.2	Бізнес-вимоги до програмного продукту	7
1.1.3	Опис проблеми споживача	7
1.1.3.1	Концептуальний опис проблеми споживача	7
1.1.3.2	Метричний опис проблеми споживача	7
1.1.4	Мета створення програмного продукту	8
1.1.5	Назва програмного продукту	9
1.2.3.1	Гасло програмного продукту	9
1.2.3.2	Логотип програмного продукту	9
1.3	Вимоги користувача до програмного продукту	10
1.1.6	Історія користувача програмного продукту	10
1.1.7	Діаграма прецедентів програмного продукту	11
1.1.8	Сценаріїв використання прецедентів програмного продукту	11
1.4	Функціональні вимоги до програмного продукту	13
1.1.9	Багаторівнева класифікація функціональних вимог. Ошибка! Закладка не определена.	
1.1.10	Функціональний аналіз існуючих програмних продуктів	14
1.5	Нефункціональні вимоги до програмного продукту	14
1.1.11	Опис зовнішніх інтерфейсів.....	14

					<i>ІКС КР 122 АІ181 ПЗ</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Студент		Горбик М. М.			Розробка веб-додатку для автоматизації діяльності кафе для працівників	Літ.	Аркуш	Аркушів
Керівник		Блажко О. А.					3	38
Н. Контр.						Пояснювальна записка		
Зав. Каф.								

1.1.11.1	Опис інтерфейса користувача.....	14
1.1.11.1.1	Опис INPUT-інтерфейса користувача.....	14
1.1.11.1.2	Опис OUTPUT-інтерфейса користувача.....	Ошибка! Закладка не определена.
1.1.11.2	Опис інтерфейсу із зовнішніми пристроями.....	15
1.1.11.3	Опис програмних інтерфейсів	16
1.1.11.4	Опис інтерфейсів передачі інформації	16
1.1.11.5	Опис атрибутів продуктивності.....	16
2.	Планування процесу розробки програмного продукту	17
2.1	Планування ітерацій розробки програмного продукту	17
2.2	Концептуальний опис архітектури програмного продукту	18
2.3	План розробки програмного продукту	18
2.3.1	Оцінка трудомісткості розробки програмного продукту.....	18
3.	Проектування програмного продукту	23
3.1	Концептуальне та логічне проектування структур даних програмного продукту	23
3.2	Проектування програмних класів	23
3.3	Проектування алгоритмів роботи методів програмних класів	24
3.4	Проектування тестових наборів методів програмних класів.....	25
4.	Конструювання програмного продукту	26
4.1	Особливості конструювання структур даних	26
4.2	Особливості конструювання програмних модулів	27
4.3	Тестування програмних модулів.....	29
5.	Розгортання та валідація програмного продукту	33
5.1	Інструкція з встановлення програмного продукту.....	33
5.2	Інструкція з використання програмного продукту	33
5.3	Результати валідації програмного продукту.....	37
	Висновки до курсової роботи	38

					ІКС КР 122 АІ181 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

1. Вимоги до програмного продукту

1.1 Визначення потреб споживача

1.1.1 Ієрархія потреб споживача

Відомо, що в теорії маркетингу потреби людини можуть бути представлені у вигляді ієрархії потреб ідей американського психолога Абрахама Маслоу включають рівні:

- фізіологія (вода, їжа, житло, сон);
- безпека (особиста, здоров'я, стабільність),
- приналежність (спілкування, дружба, любов),
- визнання (повага оточуючих, самооцінка),
- самовираження (вдосконалення, персональний розвиток).

На рисунку 1.1 представлено одну ієрархію потреби споживача, яку хотілося б задовольнити, використовуючи майбутній програмний продукт.



Рис. 1.1 – ієрархія потреб споживача

1.1.2 Деталізація матеріальної потреби

Для деталізації матеріальної потреби ми скористалися ментальними картами, що зобразили на рисунку 1.2. При створенні ментальних карт матеріальна потреба розташовувалася в центрі карти. Асоціативні гілки створювалися, припускаючи, що в загальному вигляді з об'єктом пов'язані три потоки даних/інформації: вхідний, внутрішній, вихідний. Кожен потік-це асоціативна група, що включає можливі п'ять гілок, що відповідають на п'ять питань: Хто? Що? Де? Коли? Як? Відповідно до рекомендацій щодо створення ментальних карт деякі гілки-асоціації бути розділені на додаткові асоціативні гілки, що деталізують відповіді на поставлені питання.

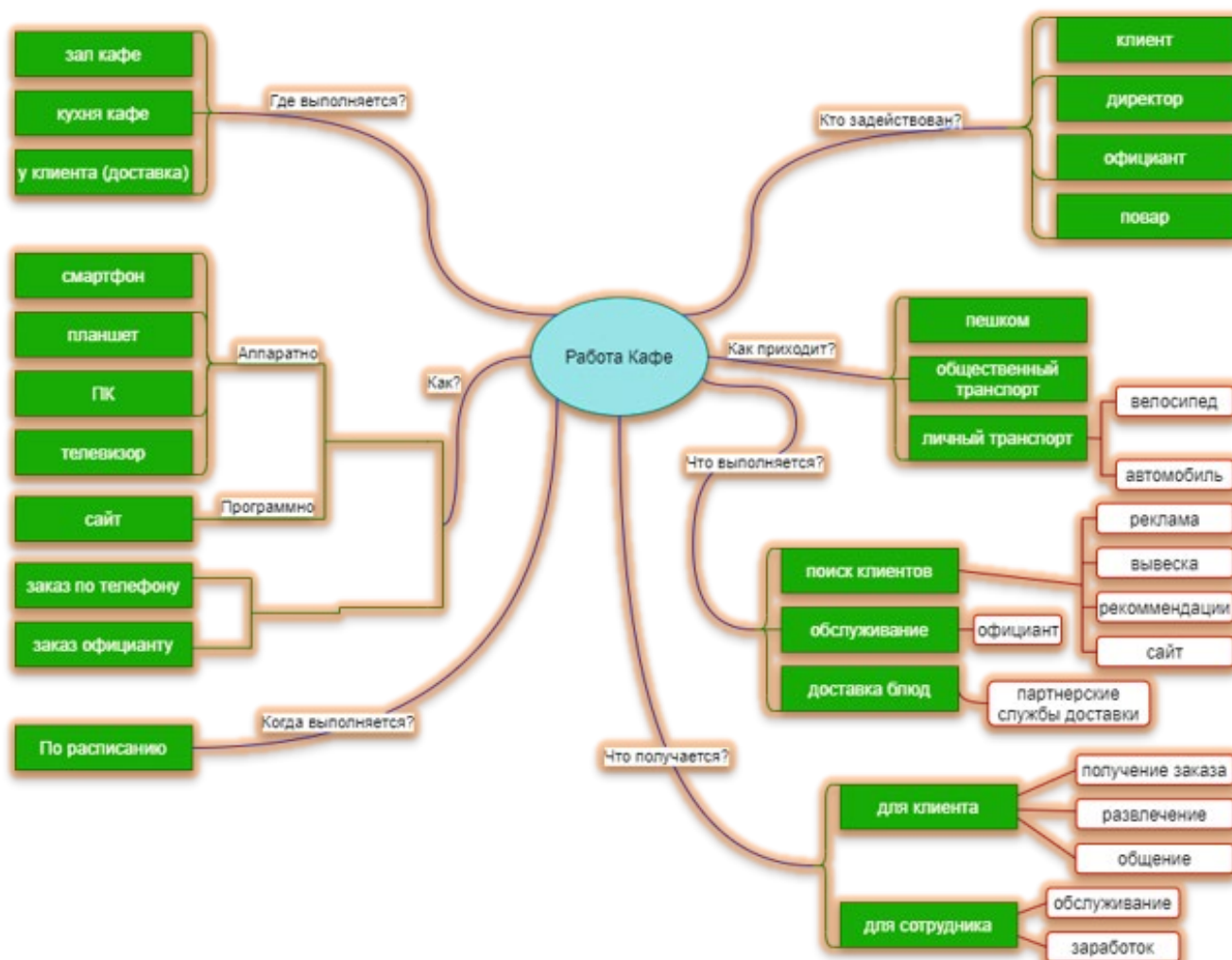


Рис. 1.2 – схема, що деталізує матеріальні потреби споживача

1.2 Бізнес-вимоги до програмного продукту

1.2.1 Опис проблеми споживача

1.2.1.1 Концептуальний опис проблеми споживача

Ціллю ПП є вирішення певних проблем, з якими стикається користувач.

Основні такі проблеми включають:

- Повільне обслуговування клієнтів через необхідність ручного запису замовлень
- Неінформованість клієнтів щодо меню і статусу роботи кафе
- Важко аналізувати дані популярності замовлень
- Можливі неточності або помилки під час вирахування вартості замовлення, загального прибутку тощо, недобропорядність

1.2.1.2 Метричний опис проблеми споживача

Відповідно до концептуального опису проблем споживача, були сформовані метричні показники незадоволеності споживача:

- Витрачений час на ручний запис замовлення, перерахунок його вартості, як результат зменшення кількості клієнтів й доходу. Середній час оформлення замовлення клієнта 1 хвилина.
- Витрати часу клієнтом на пошук необхідної інформації, збільшення часу очікування офіціантом вибору клієнта. Сердній час очікування вибору клієнта – 4 хвилини.
- Затратність по часу і зусиллям для аналізу даних щодо популярності страв. Підведення підсумків в ручному режимі в середньому займає 20 хвилин щодня.
- Потенційне зменшення прибутку через помилки ручного обчислення даних. Повторні перевірки займають в середньому 3 хвилини, без перевірок можливі втрати прибутку.

					ІКС КР 122 АІ181 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

1.2.2 Мета створення програмного продукту

1.2.2.1 Проблемний аналіз існуючих програмних продуктів

В таблиці 1.1 вказані найбільш популярні продукти які могли б бути цікавими для замовника та задовольнити його потреби, разом з примітками.

Таблиця 1.1

№	Назва продукту	Вартість	Ступінь готовності	Примітка
1	Quick Resto	платна	1	Зручна, підходить для малого та середнього бізнесу
2	Poster	платна	1	Націлена на офіціантів, може працювати оффлайн
3	NOMIA POS	Одноразова оплата+	1	Функціонал базової версії мінімальний, але можливо його розширювати підпискою на потрібні модулі
4	МойСклад	Безкошт.+	1	Більш широка сфера застосування, через що складний інтерфейс, надлишок модулів
5	YUMA	платна	1	Комплексна програма для мережі закладів

1.2.2.2 Мета створення програмного продукту

Метою створення ПП є задовільнення потреб споживача та вирішення зазначених вище проблем. Таким чином, мету можна розбити на такі пункти:

- Підвищення рівня доступності меню закладу і статусу його роботи для клієнтів– організація онлайн доступу до даних кафе, скорочення часу очікування вибору клієнта на 20% або більше.
- Підвищення швидкості обслуговування клієнтів на 20% або більше.

- Підвищення рівня доступності даних щодо популярності страв для того, щоб модернізувати меню і цінову політику – максимально знизити витрати часу на формування звітів, але хоча б на 50%.
- Підвищення точності даних прибутку й актуальності меню, виключення помилок обчислення.

1.2.3 Назва програмного продукту

1.2.3.1 Гасло програмного продукту

Назва програмного продукту:

CWOS - Cafe work organization service

(Сервіс з онлайн організації роботи кафе)

Гасло:

Ми робимо роботу персоналу простіше і швидше.

1.2.3.2 Логотип програмного продукту

Логотип ПП зображений на рисунку 1.3 та символізує швидке приготування та обслуговування.



Рис. 1.3 – логотип ПП

1.3 Вимоги користувача до програмного продукту

1.3.1 Історія користувача програмного продукту

При проектуванні ПП були виділені наступні ролі.

Клієнт – репрезентує будь-яку людину, що заходить на сайт. Клієнтом сервер за замовчанням вважає усіх хто не пройшов авторизацію. Для ролі клієнта надається право переглядати головну сторінку та сторінку меню без можливості їх редагування та якогось впливу на дані.

Офіціант – для ролі офіціанта надаються права на створення замовлення, додавання в них блюд та виключення їх, зміни кількості, а також на відміну усього замовлення. Офіціант може переглядати усі активні замовлення, час їх створення та статус готовності блюд в них.

Повар – ролі повара надаються повноваження переглядати усі наявні замовлення та «брати» їх у роботу шляхом перевизначення статусу окремих блюд. Ця операція дозволяє запобігти обираю одного і того самого блюда замовлення декількома кухарами, щоб не марнувати час і ресурси, а також дозволяє офіціантові відстежувати статус виконання замовлення, щоб знати, коли можливо подавати готове замовлення чи окреме блюдо з нього клієнтові.

Директор – роль директора включає в себе повноваження усіх інших ролей задля контролю роботи кафе і на випадок, якщо йому доведеться підмінити якогось співробітника. Крім того, директор може додавати нових співробітників до бази, змінювати дані існуючих співробітників, а також самих блюд меню. Директорові також надається можливість переглянути систематизовані дані щодо продажів усіх блюд і відповідно, прибутку з цього.

На основі цих ролей були створені історії для кожного зі співробітників, що репрезентують вигоду для кожного з них від створюваного ПП:

Клієнт ознайомлюється з меню, спеціальними пропозиціями та часом роботи кафе на сайті або в самому закладі.

Офіціант приймає замовлення клієнта та вносить його в базу даних.

Повар бачить список замовлених блюд та готує їх, докладаючи про статус приготування.

					ІКС КР 122 АІ181 ПЗ	Арк.
						11
Зм.	Арк.	№ докум.	Підпис	Дата		

Офіціант бачить статус готовності страв та відносить їх клієнтам.

Директор відслідковує звіти продажу, які формуються системою, може змінювати дані персоналу та меню.

1.3.2 Діаграма прецедентів програмного продукту

На рисунку 1.4 наведена діаграма прецедентів ПП, що ілюструє взаємодію користувачів зі створюваним сайтом.

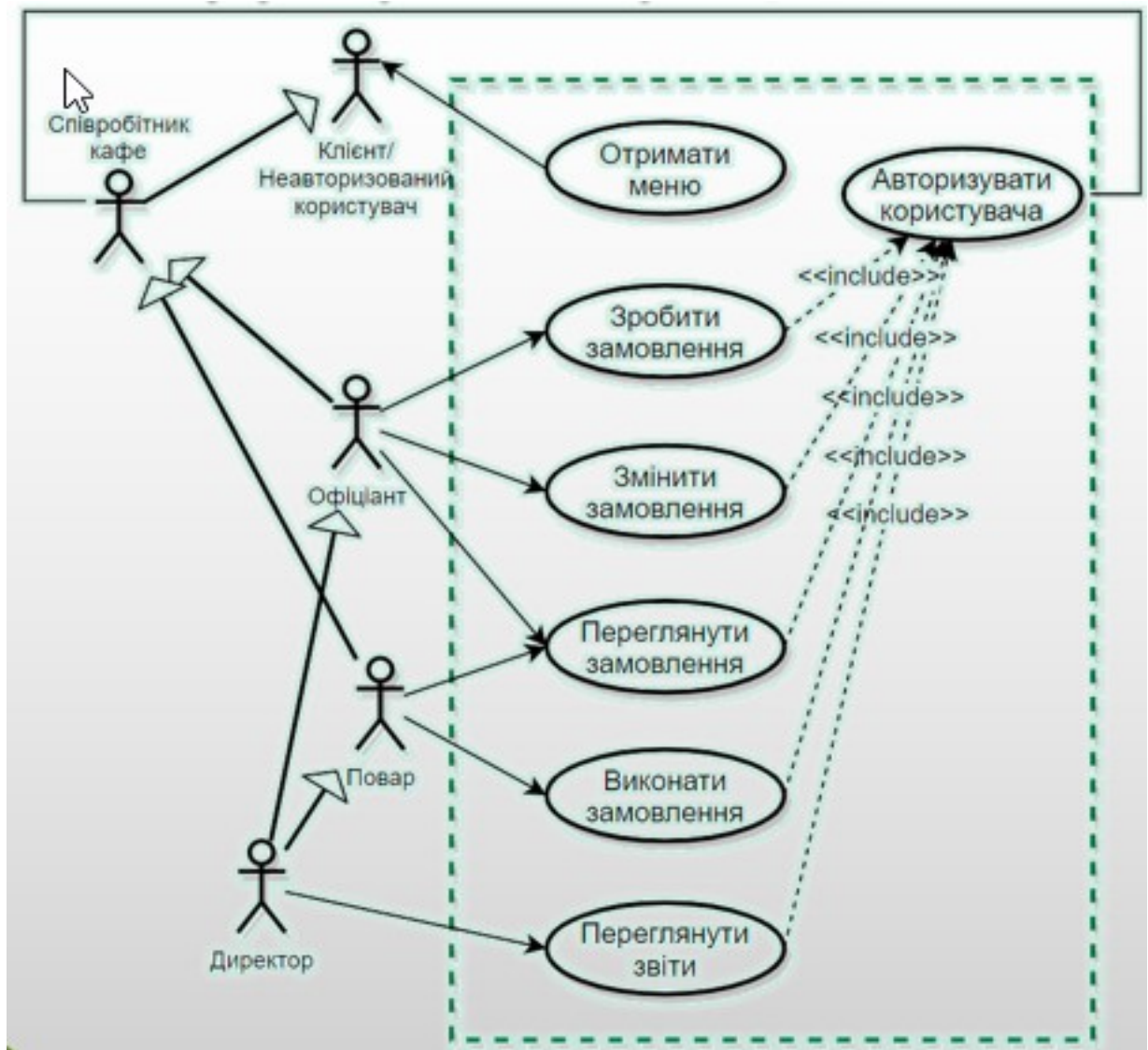


Рис. 1.4 – діаграма прецедентів

1.3.3 Сценарії використання прецедентів програмного продукту

На основі історій та діаграми прецедентів були сформовані основні сценарії використання ПП. Далі вони наведені разом с альтернативними варіантами виконання (alt).

Замовлення

1. Офіціант вказує на сторінці замовлення номер столика та можливий коментар від клієнта.

alt.1.1 ПП виявляє некоректні дані й сповіщає про помилку, повертаючи користувача до пункту 1.

2. ПП відкриває створене замовлення і офіціант додає в нього замовлені блюда у вказаній кількості.

alt.2.1 Офіціант вказує невірні дані та виявляє це після їх збереження.

alt.2.2 Офіціант видаляє некоректне замовлення або редагує його дані через відповідне меню зміни замовлення.

Зміна замовлення

1. Офіціант виконує пошук, відкриває знайдене замовлення і змінює його некоректні дані або додає/виключає з замовлення страви.

2. Підтверджує зміни.

alt.2.1 ПП виявляє некоректні дані і скасовує їх збереження, повертаючи офіціанта назад до редагування.

Авторизація

1. ПП запитує в користувача параметри авторизації.

2. Користувач вводить свої дані.

3. ПП надає доступ до сторінок сайту відповідно до виявленої ролі користувача.

alt.3.1 ПП виявляє невірність наданих даних, і повідомляє користувача.

alt.3.2 ПП повертає користувача на крок 2.

Перегляд та взаємодія

1. Співробітник може переглянути меню та замовлення.

1.1 Директор може змінити дані страв меню або виключити їх з нього за необхідності.

2. Повар/директор за потреби змінює статус готовності замовлених страв.

3. ПП відображає зміни статусу для інших співробітників.

					ІКС КР 122 АІ181 ПЗ	Арк.
						13
Зм.	Арк.	№ докум.	Підпис	Дата		

1.4 Функціональні вимоги до програмного продукту

1.4.1 Багаторівнева класифікація функціональних вимог

На основі сценаріїв використання були спроектовані функції, що наведені в таблиці 1.2

Таблиця 1.2 – виділенням позначене моє завдання

Ідентифікатор	Назва функції
FR1	Авторизація
FR1.1	Створення продуктом запиту на параметри ідентифікації користувача.
FR1.2	Передача даних від користувача й надання йому повноважень ПП згідно ролі офіціанта/кухара/директора
FR1.2_alt	Виявлення ПП некоректності даних, сповіщення про помилку, - >1.1
FR2	Опрацювання нового замовлення
FR2.1	ПП створює форму для введення даних столика та можливого коментаря від клієнта
FR2.2	Офіціант вказує на сторінці замовлення номер столика та можливий коментар від клієнта, підтверджує
FR2.3	ПП зберігає коректне замовлення в спільному списку для співробітників і відображає його.
FR2.4	Офіціант додає в замовлення страви у вказаній кількості, підтверджує і ПП оновлює дані замовлення.
FR2.2,2.4_alt	Виявлення ПП некоректності даних, сповіщення про помилку, повтор запиту даних
FR3	Зміна замовлення
FR3.1	Офіціант знаходить необхідне замовлення і відкриває його, змінює некоректні дані або додає/виключає з замовлення страви. Для кухара зміна замовлення передбачає зміну статусу готовності окремих страв
FR3.2	Співробітник підтверджує зміни, програма вносить їх в базу даних
FR3.2_alt	ПП виявляє некоректні дані і скасовує їх збереження, повертаючи офіціанта назад до редагування
FR4	Менеджмент даних кафе
FR4.1	Директор переглядає та редагує за необхідності дані співробітників або меню, додаючи, змінюючи або видаляючи елементи й підтверджує зміни.
FR4.2	ПП перевіряє дані та застосовує їх, або повертає на виправлення.
FR5	Перегляд звітів
FR5.1	Директор виконує запит на перегляд звітів
FR5.2	ПП формує звіти на основі останніх даних та надає їх директору

1.4.2 Функціональний аналіз існуючих програмних продуктів

Відповідно до спроектованих функцій було проведено аналіз схожих ПП, дані порівняння були відображені в таблиці 1.3

Таблиця 1.3

Ідентифікатор	Quick Resto	Poster	NOMIA POS*	МойСклад	YUMA
FR 1.1	+	+	+	+	+
FR 1.2	+	+	+	-	+
FR 2.1	+	+	+	~-	+
FR 2.2	+	+	+	+	+
FR 3.1	+	+	+	+	+
FR 3.2	+	+	+	+	+
FR 4.1	~+	+	-	-	-
FR 4.2	+	+	-	-	-
FR 5.1	+	+	-	+	+
FR 5.2	+	+	-	+	+

1.5 Нефункціональні вимоги до програмного продукту

1.5.1 Опис зовнішніх інтерфейсів

1.5.1.1 Опис інтерфейса користувача

1.5.1.1.1 Опис INPUT-інтерфейса користувача

Input-інтерфейс користувача описано в таблиці 1.4

Таблиця 1.4

Ідентифікатор	Засіб INPUT-потoku	Особливості використання	Зовнішній пристрій
FR 1	Стандартна клавіатура, миша / сенсорний екран	Можна підтвердити введення клавішею «enter», або натиснути екранну кнопку.	Смартфон/планшет, ноутбук, настільний комп'ютер
FR 2	Стандартна клавіатура, миша / сенсорний екран		
FR 3	Стандартна клавіатура, миша / сенсорний екран	Статус готовності змінюється натисканням відповідних екранних кнопок	

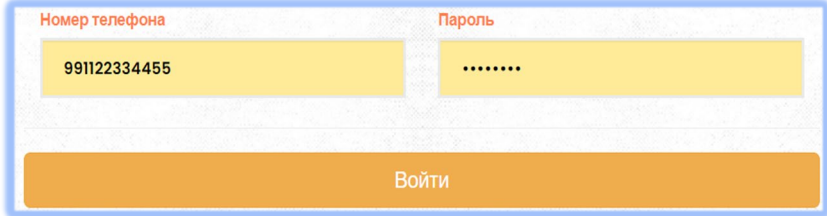
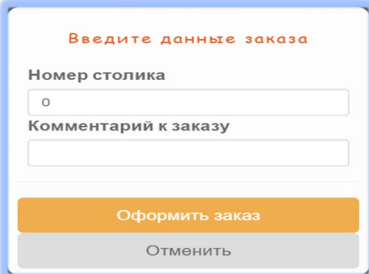
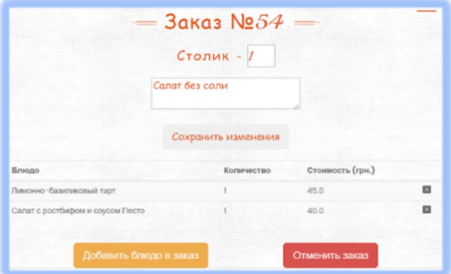
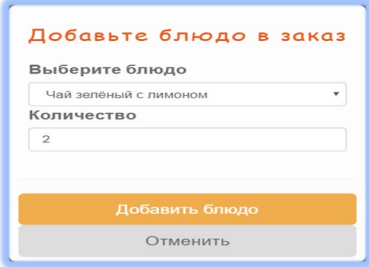
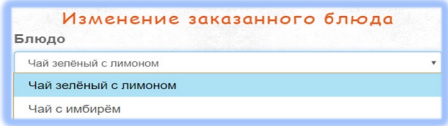
Таблиця 1.4 (продовження)

FR 4	Стандартна клавіатура, миша / сенсорний екран	Можна підтвердити введення клавішею «enter», або натиснути екранну Кнопку.	
FR 5	Миша / сенсорний екран		

1.5.1.1.2 Опис OUTPUT-інтерфейса користувача

Output-інтерфейс користувача із мокапами, тобто орієнтовними зображеннями описано в таблиці 1.5

Таблиця 1.5

Ідентифікатор	Засіб OUTPUT-поток	Особливості використання
FR 1	Графічний інтерфейс	
FR 2		
FR 3		  

1.5.1.2 Опис інтерфейсу із зовнішніми пристроями

Усі функції ПП належним чином мають працювати незалежно від того, яким пристроєм буде користуватися клієнт, тобто опис інтерфейсу можна представити таблицею 1.6

					ІКС КР 122 АІ181 ПЗ	Арк.
						16
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 1.6

Ідентифікатор	Зовнішній пристрій
FR 1	Смартфон/ планшет, ноутбук (з маніпулятором або сенсорним екраном), настільний комп'ютер (з мишею та клавіатурою)
FR 2	
FR 3	
FR 4	
FR 5	

1.5.1.3 Опис програмних інтерфейсів

Для коректної роботи ППі підходить будь-яка сучасна оновлена операційна система та браузер для клієнта; Linux та Spring MVC для сервера; та PostgreSQL для Бази Даних.

1.5.1.4 Опис інтерфейсів передачі інформації

Використовуються наступні інтерфейси передачі даних:

Для ПК:

- Ethernet

Для ноутбуків:

- WiFi

Для смартфонів/планшетів:

-WiFi

-Сотова мережа (3G/LTE/...++)

1.5.1.5 Опис атрибутів продуктивності

Бажаний та допустимий для замовника максимальний час реакції ППі на дії користувача зазначено у таблиці 1.7.

Ідентифікатор	Максимальний час реакції ППі на дії користувача в секундах
FR1.1	5
FR1.2	5
FR2.1	1
FR2.2	1
FR2.2 alt	3
FR3.1	3
FR3.2/alt	1
FR4.1	5
FR4.2	5
FR5.1	1
FR5.2	5

2. Планування процесу розробки програмного продукту

2.1 Планування ітерацій розробки програмного продукту

З метою забезпечення для вимог таких рекомендацій IEEE-стандарту, як необхідність, корисність при експлуатації, здійсненність функціональних вимог до ПП, були визначені функціональні пріоритети, які будуть використані при плануванні ітерацій розробки ПП. Вони позначені нижче в таблиці 2.1.

Таблиця 2.1

Ідентифікатор функції	Функціональні Залежності	Вплив на досягнення мети, %	Пріоритет функції
FR1 - Авторизація	FR4	0	M
FR1.1	-	0	M
FR1.2	FR4	0	M
FR1.2_alt	FR4	0	M
FR2 – створення замовлення	FR1	32	S
FR2.1	-	8	S
FR2.2	FR1	8	S
FR2.3	-	8	S
FR2.4	FR1	8	S
FR2.2,2.4_alt	FR2.1	0	S
FR3 – зміна замовлень	FR1, FR2	40	S
FR3.1	FR1, FR2	30	S
FR3.2	FR1, FR2	5	S
FR3.2_alt	-	5	S
FR4 – менеджмент співробітників	FR1	14	M
FR4.1	FR1	13	M
FR4.2	-	1	C
FR5 -звіти	FR1, FR2	14	W
FR5.1	FR1	1	W
FR5.2	FR2	13	W

2.2 Концептуальний опис архітектури програмного продукту

Архітектура програмного продукту включає Presentation level, тобто інтерфейс, Business level, тобто серверну частину, та Access level, що представляє БД та взаємодію з БД. Такий опис може бути відображений на рисунку 2.1.

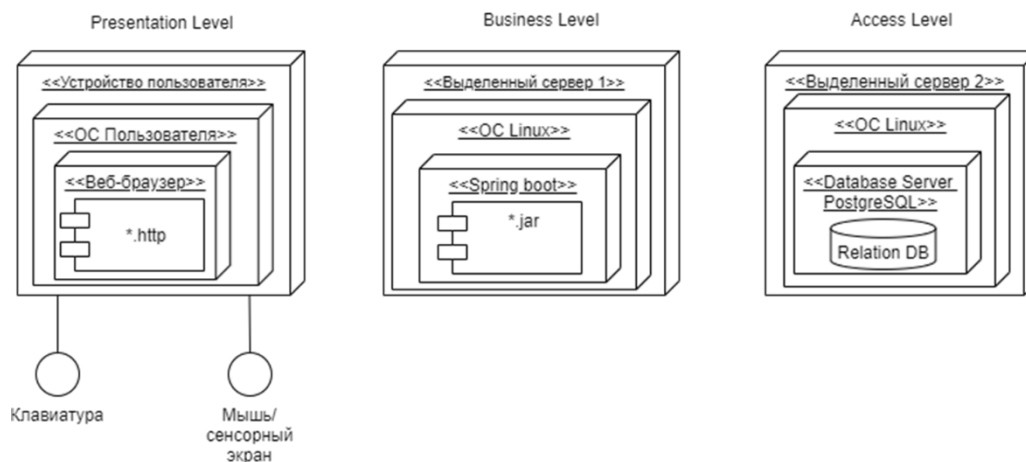


Рис. 2.1 – діаграма архітектури

2.3 План розробки програмного продукту

2.3.1 Оцінка трудомісткості розробки програмного продукту

Оцінка трудомісткості процесу проводилася за допомогою так званих Use Case Points. Нижче будуть наведені таблиці для визначення усіх необхідних для цього показників.

Таблица 2.2 - Вагові коефіцієнти

Тип актора	Ваговий коефіцієнт	Наявна кількість
Простий	1	0
Середній	2	0
Складний	3	3

Таблица 2.3

Тип прецедента	Кількість кроків сценарію	Ваговий коефіцієнт	Наявна кількість
Простий	≤ 3	5	4
Середній	4-7	10	1
Складний	> 7	15	0

Сценарії використання прецедентів ПП

$$UUCP = 3*3 + (4*5 + 1*10) = 39$$

Таблиця 2.4 - Сценарії використання прецедентів ПП

Показник	Опис показника	Вага	STi
T1	Розподілена система	2	2
T2	Висока продуктивність (пропускна здатність)	1	1
T3	Робота кінцевих користувачів в режимі он-лайн	1	5
T4	Складна обробка даних	-1	0
T5	Повторне використання коду	1	5
T6	Простота установки	0,5	5
T7	Простота використання	0,5	5
T8	Переносимість	2	5
T9	Простота внесення змін	1	3
T10	Паралелізм	1	3
T11	Спеціальні вимоги до безпеки	1	2
T12	Безпосередній доступ до системи з боку зовнішніх користувачів	1	0
T13	Спеціальні вимоги до навчання	1	0

Визначення рівня кваліфікації розробників

$$TCF = 0,6 + (0,01 * \sum(STi * Вага_i)) = 0,6 + 0,35 = 0,95$$

Таблиця 2.5 - Визначення рівня кваліфікації розробників

Показник	Опис показника	Вага	Значення
F1	Знайомство з технологією	2	3
F2	Досвід розробки додатків	1	2
F3	Досвід використання об'єктно-орієнтованого підходу	1	3
F4	Наявність ведучого аналітика	-1	0
F5	Мотивованість	1	3
F6	Стабільність вимог	0,5	2
F7	Часткова зайнятість	0,5	5
F8	Складні мови програмування	2	3

$$EF = 1,4 + (-0,03 * \sum(STi * Вага_i)) = 1,4 - 0,03 * 25,5 = 0,635$$

$$Use\ Case\ Points = 39 * 0,95 * 0,635 = 23,5 - \text{прийнятний показник}$$

2.3.2 Визначення дерева робіт з розробки програмного продукту

Дерево розробки може бути представлено наступним чином:

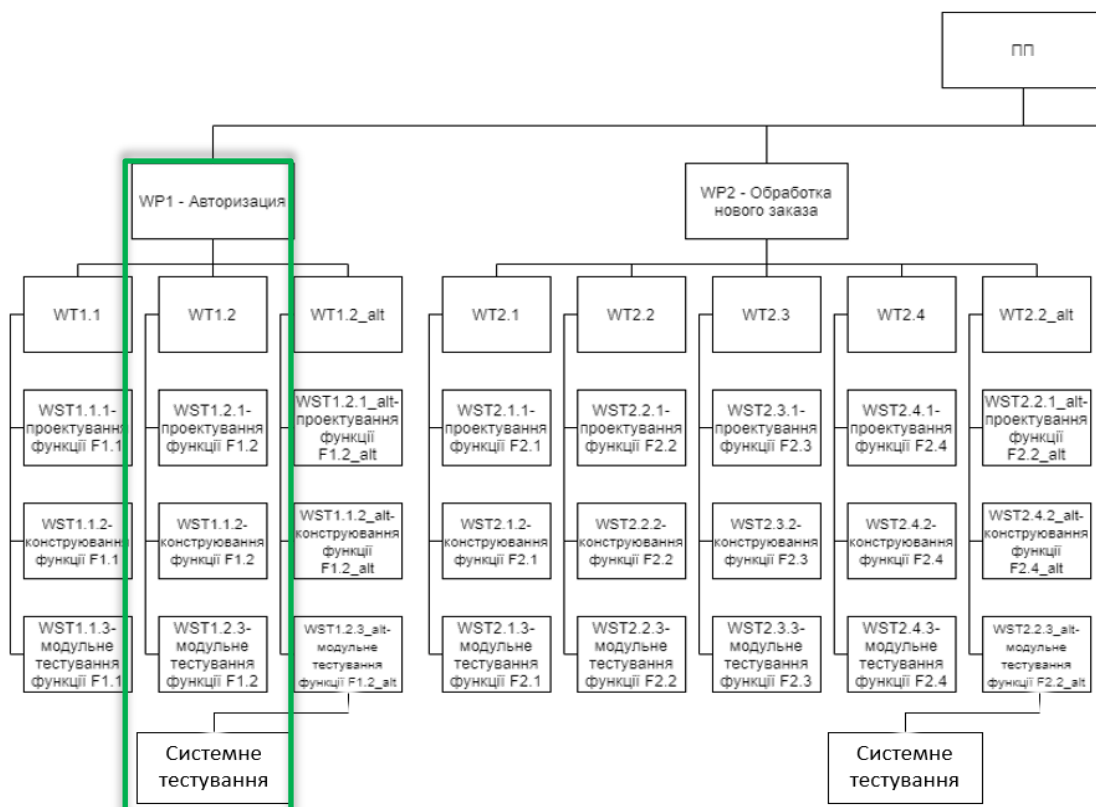


Рис. 3.2 – дерево розробки (функції F1, F2) – рамкою позначене моє завдання

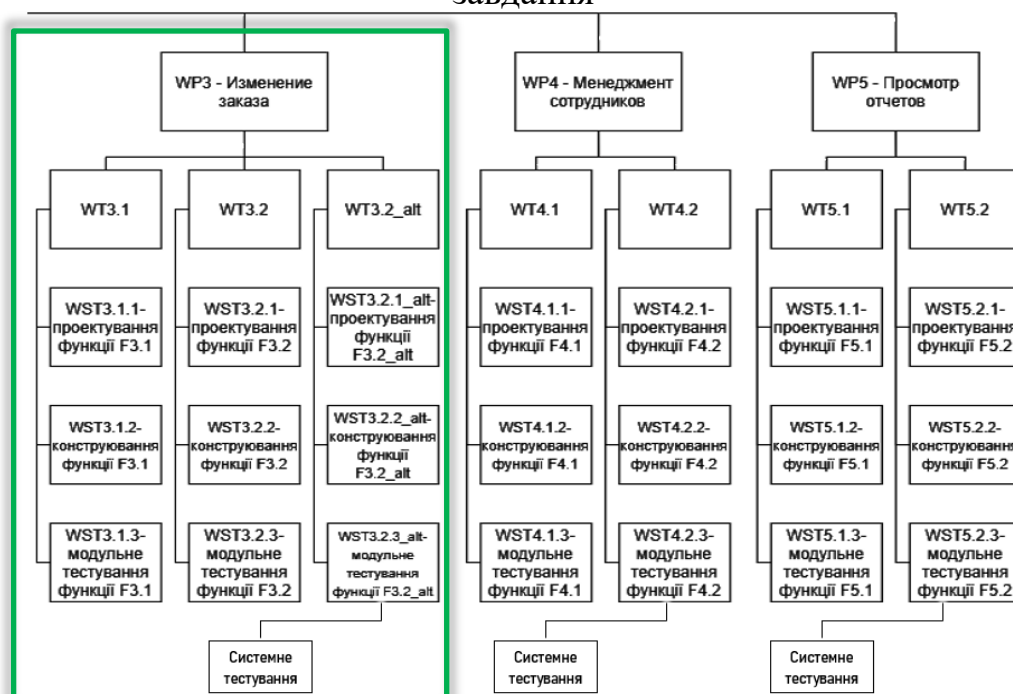


Рис. 4.3 – дерево розробки (функції F3, F4, F5) – рамкою позначене моє завдання

Зм.	Арк.	№ докум.	Підпис	Дата

Таблиця 2.5 – закріплення підзадач за виконавцями

WST	Виконавець
1.1.1	Архіпов Д.
1.1.2	Архіпов Д.
1.1.3	Архіпов Д.
1.2.1	Горбик М.
1.2.2	Горбик М.
1.2.3	Горбик М.
1.2.1 alt	Архіпов Д.
1.2.2 alt	Архіпов Д.
1.2.3 alt	Архіпов Д.
2.1.1	Козиревич Я.
2.1.2	Козиревич Я.
2.1.3	Козиревич Я.
2.2.1	Козиревич Я.
2.2.2	Козиревич Я.
2.2.3	Козиревич Я.
2.3.1	Козиревич Я.
2.3.2	Козиревич Я.
2.3.3	Козиревич Я.
2.4.1	Козиревич Я.
2.4.2	Козиревич Я.
2.4.3	Козиревич Я.
2.2.1 alt	Козиревич Я.
2.2.2 alt	Козиревич Я.
2.2.3 alt	Козиревич Я.
3.1.1	Горбик М.
3.1.2	Горбик М.
3.1.3	Горбик М.
3.2.1	Горбик М.
3.2.2	Горбик М.
3.2.3	Горбик М.
3.2.1 alt	Горбик М.
3.2.2 alt	Горбик М.
3.2.3 alt	Горбик М.
4.1.1	Архіпов Д.
4.1.2	Архіпов Д.
4.1.3	Архіпов Д.
4.2.1	Архіпов Д.
4.2.2	Архіпов Д.
4.2.3	Архіпов Д.
5.1.1	Архіпов Д.
5.1.2	Архіпов Д.
5.1.3	Архіпов Д.
5.2.1	Архіпов Д.
5.2.2	Архіпов Д.
5.2.3	Архіпов Д.

2.3.3 Графік робіт з розробки програмного продукту

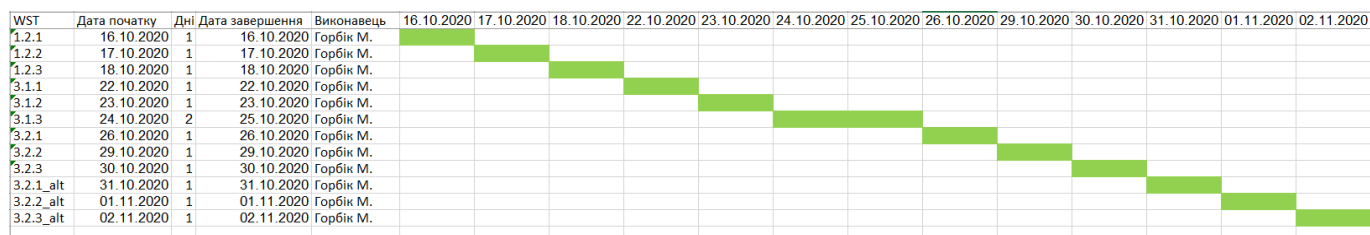
2.3.3.1 Таблиця з графіком робіт

Графік реалізації мною функцій зазначений в таблиці нижче

Таблиця 2.6

WST	Дата початку	Дні	Дата заверше	Виконавець
1.2.1	16.10.2020	1	16.10.2020	Горбик М.
1.2.2	17.10.2020	1	17.10.2020	Горбик М.
1.2.3	18.10.2020	1	18.10.2020	Горбик М.
3.1.1	22.10.2020	1	22.10.2020	Горбик М.
3.1.2	23.10.2020	1	23.10.2020	Горбик М.
3.1.3	24.10.2020	2	25.10.2020	Горбик М.
3.2.1	26.10.2020	1	26.10.2020	Горбик М.
3.2.2	29.10.2020	1	29.10.2020	Горбик М.
3.2.3	30.10.2020	1	30.10.2020	Горбик М.
3.2.1_alt	31.10.2020	1	31.10.2020	Горбик М.
3.2.2_alt	01.11.2020	1	01.11.2020	Горбик М.
3.2.3_alt	02.11.2020	1	02.11.2020	Горбик М.

2.3.3.2 Діаграма Ганта



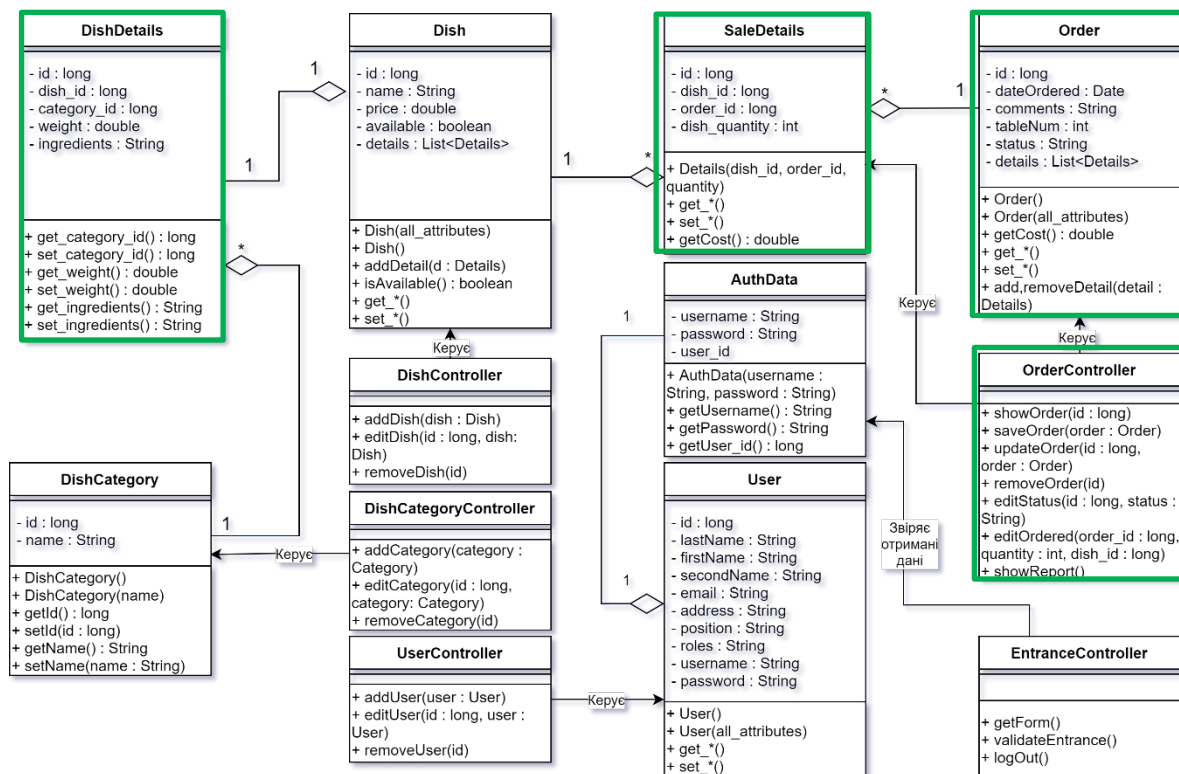


Рис. 3.2 - UML-діаграми програмних класів (рамкою виділено мою частину)

3.3 Проектування алгоритмів роботи методів програмних класів

На рисунках 3.3 – 3.7 показано UML-діаграми активності роботи відповідних класів.

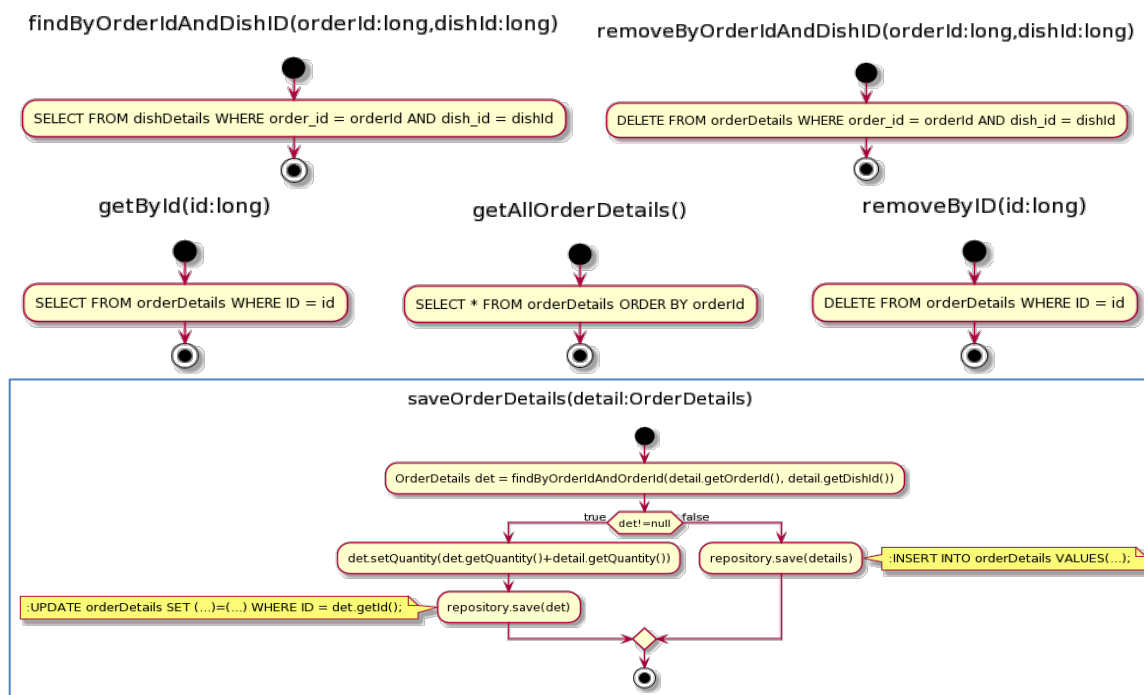


Рис. 3.3 - UML-діаграма активності роботи класу OrderDetails

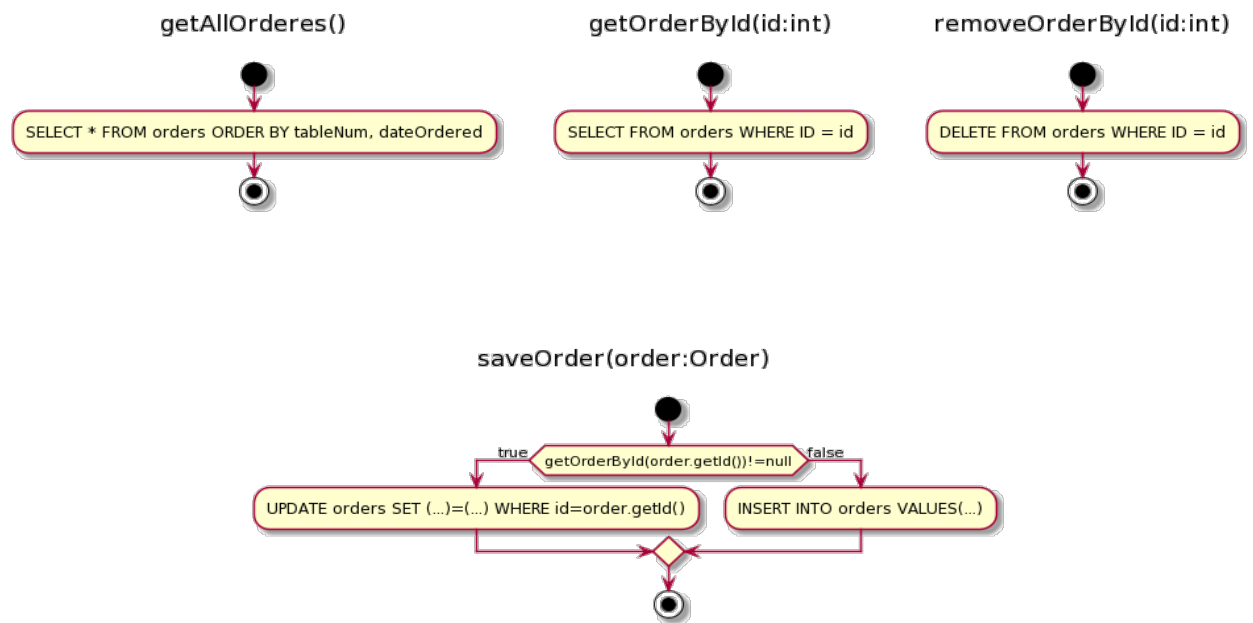


Рис. 3.4 - UML-діаграма активності роботи класу Order

3.4 Проектування тестових наборів методів програмних класів

В таблиці 3.1 відображено тестові набори програмних класів.

Таблиця 3.1

Назва функції	№ тесту	Опис значень вхідних даних	Опис очікуваних значень результату
sortByQuantity	1	List<Details>: {[1,3],[2,4],[3,1]}	List<Details>: {[1,4],[2,3],[3,1]}
removeFromOrder(id)	1	Задано: List<Details>: {[1,3],[2,4],[3,1]} Перевіряємо (2)	Список набуває вигляду List<Details>: {[1,3],[3,1]}
	2	Задано: List<Details>: {[1,3],[2,4],[3,1]} Перевіряємо (f)	NoSuchElementException

4. Конструювання програмного продукту

4.1 Особливості конструювання структур даних

4.1.1 Особливості інсталяції та роботи з СУБД

Використана СУБД – PostgreSQL 13.1 / Heroku Postgres;

Інсталяція БД проводилася на локальні комп'ютери задля незалежного тестування. Для клієнта ПП такої потреби немає, оскільки база знаходиться на сервері Heroku. Робота з СУБД для клієнта відбувається лише засобами графічного інтерфейсу в браузері та не потребує встановлення додаткового програмного або апаратного забезпечення.

4.1.2 Особливості створення структур даних

Після налаштування підключення до БД, запити створення таблиць формуються автоматично за допомогою Hibernate та за необхідності таблиці оновлюються при зміні програмного коду – також автоматично. Hibernate орієнтується на анотації типу @Table, @Entity, @Id, @OneToMany тощо, які ми додаємо на етапі написання класів. Нижче зазначені класи для мого варіанта.

```
@Entity
@Table(name = "orders")
@Data
public class Order {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private long id;
    private Date date_ordered;
    private String comments;
    private int table_num;
    private String status;

    @OneToMany(mappedBy = "order", cascade = CascadeType.ALL)
    private List<Details> details;
```

Рис. 4.5 - налаштування роботи Hibernate для класу Order

```

@Entity
@Table(name = "details")
@Data
public class Details {
    @Id
    @GeneratedValue(strategy = GenerationType.AUT0)
    private long id;
    @Column(insertable = false, updatable = false)
    private long dish_id = -1;
    @Column(insertable = false, updatable = false)
    private long order_id = -1;
    private int quantity;

    @ManyToOne(optional = false, cascade = CascadeType.ALL)
    @JoinColumn(name = "dish_id")
    private Dish dish;

    @ManyToOne(optional = false, cascade = CascadeType.ALL)
    @JoinColumn(name = "order_id")
    private Order order;
}

```

Рис. 4.2 - налаштування роботи Hibernate для класу Details

4.2 Особливості конструювання програмних модулів

4.2.1 Особливості роботи з інтегрованим середовищем розробки

Використовувалося середовище IntelliJ IDEA Ultimate – зручне, швидке та знайоме; інсталяція проводилася з офіційного сайту (jetbrains.com), ліцензія – студентська.

Використовувалися фреймворки Spring (+Spring Security, Data), Bootstrap, Maven – для значного пришвидшення написання програми та уникнення дублювання коду.

4.2.2 Особливості створення програмної структури з урахуванням спеціалізованого Фреймворку

Для організації роботи у Spring та відповідно до вимог Spring Data, для зв'язку класів програми, інтерфейсу користувача та бази даних були створені контролери, сервіси та репозиторії.

4.2.3 Особливості створення програмних класів

Оскільки деякі програмні класи водночас є сутностями, що будуть представлені в таблиці БД, їх програмування велося з урахуванням цього аспекту, а саме: додавалися спеціальні поля типу `id` та колекції об'єктів, що генеруються або передаються з БД та відповідні анотації, що організовували зв'язки між сутностями-класами в програмі та відповідно таблицями у БД.

Для уникнення написання зайвого шаблонного коду також використовувався плагін Lombok, який за анотаціями типу `@Data` створював необхідні конструктори, гетери та сетери.

```
public void sortByQuantity() {
    details.sort(Comparator.comparingInt(Details::getQuantity).reversed());
}

public Details getDetailsIfPresent(long id) {
    return details.stream().filter(i -> i.getDish_id() == id).findFirst().orElse( other: null);
}

public boolean removeFromOrder(long id) {
    Details detail = details.stream().filter(i -> i.getDish().getId() == id
        || i.getDish_id() == id).findFirst().orElse( other: null);
    if (detail == null)
        throw new NoSuchElementException();
    System.out.println("Удаляем из заказа №" + id + ": " + detail);
    return details.remove(detail);
}

public void removeDetail(Details detail) {
    details.remove(detail);
}
```

Рис. 4.3 – функції, реалізовані в класі Order, відсутність гетерів та сетерів завдяки наявності Lombok

```
public long getDish_id() {
    if (dish == null)
        return dish_id;
    return dish.getId();
}

public int getQuantity() { return quantity; }

public void setQuantity(int quantity) {
    this.quantity = quantity;
}

public double getCost() {
    return dish.getPrice() * quantity;
}
```

Рис. 4.4 – функції, реалізовані в класі Details, відсутність гетерів та сетерів завдяки наявності Lombok

4.2.4 Особливості розробки алгоритмів методів програмних класів або процедур/функцій

Додатково, для налагодження інтерфейсу користувача та забезпечення повного функціоналу, використовувалася мова JavaScript та її бібліотека JQuery.

4.3 Тестування програмних модулів

Для тестування використовувалося середовище розробки IntelliJ IDEA Ultimate та фреймворк автоматичного тестування Junit 5. Написання тестів велося на основі проекту тестових наборів, описаних в лабораторній роботі №8.

Створення тестів велося у спеціальній директорії проекту з назвами, що відображають клас, що тестується типу OrderTest. Кожен окремий набір тестових даних виділявся анотацією @Test та мав включати очікувані та реальні дані, на основі порівняння яких вирішувалося, чи був тест пройдений.

Функції для тестування обиралися виходячи з найбільшої вірогідності виникнення в них помилки.

Специфікація функції sortByQuantity класу Order має наступний опис.

Вхідні параметри відсутні.

Значення не повертаються функцією. Для перевірки коректності роботи, викликається функція getDishNames.

					ІКС КР 122 АІ181 ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4 – Опис тестових наборів для функції sortByQuantity

№ тесту		Опис значень початкових даних	Опис очікуваних значень результату
6	Поточні дані - Випадкова кількість замовлених страв від 1 до X Наприклад: Кава (2 од.) , Тістечко (1 од.), Печиво (3 од.)	Відсортовані елементи за спадом Наприклад: Кава (2 од.) , Тістечко (1 од.), Печиво (3 од.)	

```

class OrderTest {

    int iterations = 25;

    @Test
    void sortByQuantity() {
        Order order = new Order();
        List<Details> details = new ArrayList<>();
        List<Integer> expectedList = new ArrayList<>();
        int[] actual = new int[iterations];
        int[] expected = new int[iterations];

        for (int i = 0; i < iterations; i++) {
            int randQuantity = ThreadLocalRandom.current().nextInt( origin: 1, bound: 10);
            details.add(new Details(i, order_id: 1, randQuantity));
            expectedList.add(randQuantity);
        }

        expectedList.sort(Collections.reverseOrder());
        order.setDetails(details);
        order.sortByQuantity();
        List<Details> temp = order.getDetails();
        for (int i = 0; i < iterations; i++) {
            actual[i] = temp.get(i).getQuantity();
            expected[i] = expectedList.get(i);
        }
        StringBuilder actualStr = new StringBuilder();
        StringBuilder expectedStr = new StringBuilder();
        for (int i = 0; i < iterations; i++) {
            actualStr.append(actual[i]);
            expectedStr.append(expected[i]);
        }
        System.out.println("Actual: " + actualStr + "\nExpected: " + expectedStr);
        assertEquals(expected, actual);
    }
}

```

Рис.4. 9 – програмний код тесту 6

✓ Test Results	34 ms	"C:\Program Files\Java\jdk-15\bin\java.exe" ...
✓ OrderTest	34 ms	Actual quantities list: 8887766666555443332222211
✓ sortByQuantity()	34 ms	Expected quantities list: 8887766666555443332222211

Рис. 4.10 – результат тесту 6

Вхідні параметри відсутні.

Значення, що повертаються функцією:

число з плаваючою комою – сума добутків усієї кількості замовлених страв на їх вартості;

					ІКС КР 122 АІ181 ПЗ	Арк.
						31
Зм.	Арк.	№ докум.	Підпис	Дата		

Специфікація функції `removeFromOrder` класу `Order` має наступний опис.

Вхідні параметри:

`id` – ідентифікатор страви, що має бути видаленою з замовлення (ціле число)

```
class OrderTest {
    @Test
    void getCost() {
        Dish dish1 = new Dish( name: "dish1", price: 50);
        Dish dish2 = new Dish( name: "dish2", price: 100);
        Dish dish3 = new Dish( name: "dish3", price: 10);
        Details details1 = new Details(dish1, quantity: 2);
        Details details2 = new Details(dish2, quantity: 1);
        Details details3 = new Details(dish3, quantity: 2);
        Order order = new Order(new ArrayList<>(Arrays.asList(details1, details2, details3)));
        double actual = order.getCost();
        double expected = 220;
        assertEquals(expected, actual);
    }
}
```

Рис. 4.13 – результат тесту 8

Значення, що повертаються функцією відстуні, окрім випадку відсутності видаляемого елемента – в такому разі виникає виняток - `NoSuchElementException`. Для перевірки коректності роботи, викликається функція `getDishNames`.

Таблиця 7 – Опис тестових наборів для функції `removeFromOrder` класу `Order`

№ тесту	Опис значень вхідних даних	Опис очікуваних значень результату
9	Випадкові дані - наприклад Кава (2 од.) [id=1], Тістечко (1 од.) [id=2], Печиво (3 од.) [id=3] id=2	Кава (2 од.), Печиво (3 од.)
10	Випадкові дані - наприклад Кава (2 од.) [id=1], Печиво (3 од.) [id=3] id=2	<code>NoSuchElementException</code>

```

@Test
void removeFromOrder() {
    Order order = new Order();
    List<Details> details = new ArrayList<>();
    List<Details> expectedList = new ArrayList<>();
    for (int i = 0; i < iterations; i++) {
        int randQuantity = ThreadLocalRandom.current().nextInt( origin: 1, bound: 11);
        //System.out.print(randQuantity+" ");
        details.add(new Details(i, order_id: 1, randQuantity, new Dish(i)));
        if (i != 5)
            expectedList.add(new Details(i, i, randQuantity, new Dish(i)));
    }
    order.setDetails(details);
    order.removeFromOrder( id: 5);
    long[] actual = new long[iterations];
    long[] expected = new long[iterations];
    List<Details> temp = order.getDetails();
    for (int i = 0; i < iterations-1; i++) {
        actual[i] = temp.get(i).getId();
        expected[i] = expectedList.get(i).getId();
    }
    System.out.println(expectedList.equals(order.getDetails()));
    assertEquals(actual, expected);
}

```

Рис. 4.94 – программный код тесту 9

✓ Test Results	35 ms	"C:\Program Files\Java\jdk-15\bin\java.exe" ...
✓ OrderTest	35 ms	Удаляем из заказа №1: Details{dish_id=1, order_id=1, id=0, quantity=4}
✓ removeFromOrder()	35 ms	[Details{dish_id=0, order_id=1, id=0, quantity=10}, Details{dish_id=2, order_id=1, id=0, quantity=9}, Details{dish_id=3,

Рис. 4.85 – результат тесту 9

```

@Test
void removeFromOrderNotExisting() {
    Order order = new Order();
    List<Details> details = new ArrayList<>();
    List<Details> expectedList = new ArrayList<>();
    for (int i = 0; i < iterations; i++) {
        int randQuantity = ThreadLocalRandom.current().nextInt( origin: 1, bound: 11);
        details.add(new Details(i, order_id: 1, randQuantity, new Dish(i)));
    }
    order.setDetails(details);
    assertThrows(NoSuchElementException.class, ()->order.removeFromOrder( id: 35));
    try {
        order.removeFromOrder( id: 35);
    }catch (NoSuchElementException e){
        System.out.println("Исключение выявлено.");
    }
}

```

Рис. 4.76 – программный код тесту 10

✓ Test Results	37 ms	"C:\Program Files\Java\jdk-15\bin\java.exe" ...
✓ OrderTest	37 ms	Исключение выявлено.
✓ removeFromOrderNotExisting()	37 ms	

Рис. 4.67 – результат тесту 10

5. Розгортання та валідація програмного продукту

5.1 Інструкція з встановлення програмного продукту

Безпосередньо інсталяція програмного продукту на пристрій користувача не проводиться, оскільки ПП представляє собою веб-сайт. Для взаємодії з сайтом, необхідними будуть відповідні зовнішні пристрої, що були зазначені в пункті 1.5.1.2 цієї записки (будь-який з перелічених), також користувачу необхідно мати актуальний оновлений браузер на своєму пристрої, наприклад Google Chrome версії 83.0.4103.97 та вище, а також підключення до мережі інтернет (бажано зазначене в п.1.5.1.4). Почати роботу з ПП користувач зможе перейшовши за посиланням <http://cafe-oop.herokuapp.com>

5.2 Інструкція з використання програмного продукту

5.2.1 Авторизація

При переході на сайт, користувачеві необхідно пройти авторизацію. Для цього на головній сторінці, потрібно вибрати пункт «Вход» (рисунок 5.1)

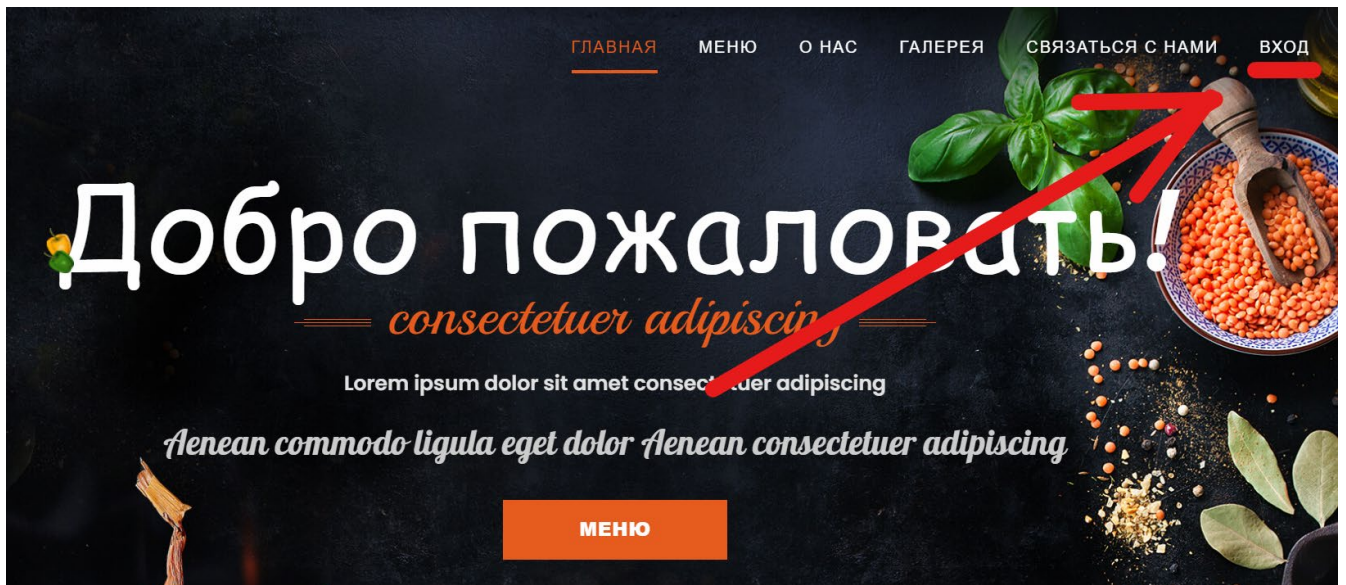


Рис. 5.1 – головна сторінка сайту ПП

Після цього з'явиться сторінка з формою авторизації (рисунок 5.2). Клієнт має ввести власні дані, що він отримав від директора. Якщо вхід виконує директор та це перший його вхід, він може запитати дані для входу від постачальника ПП, або перейти за секретним посиланням та ввести дані згенерованого початкового користувача з функціоналом директора.

					ІКС КР 122 АІ181 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

Після коректної авторизації, ПП автоматично перенаправить користувача на початкову сторінку згідно з його правами.

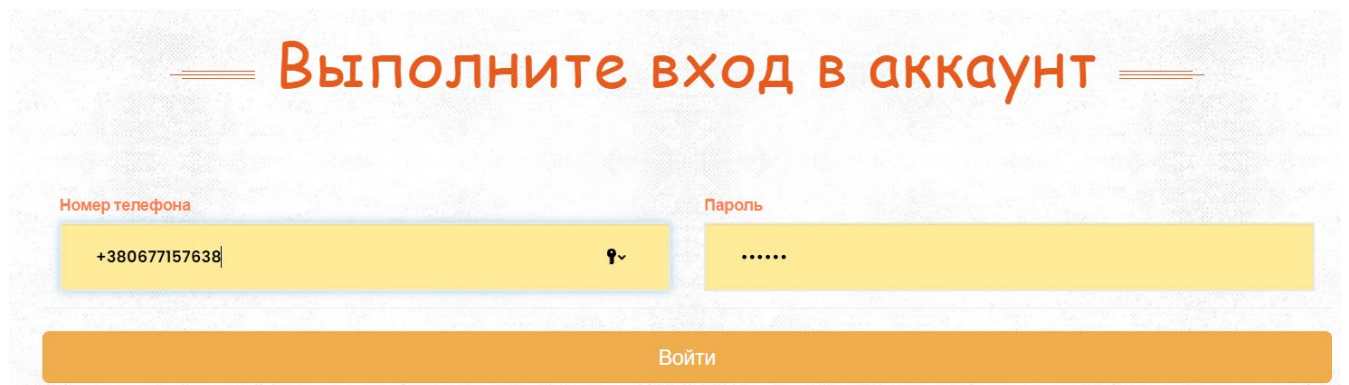


Рис. 5.2 –сторінка авторизації

5.2.2 Перегляд та створення замовлень

Після авторизації для офіціанта та директора стане доступною можливість переглянути поточні замовлення, зробити нове чи змінити існуюче замовлення. Після переходу за вкладкою «Заказы», клієнтові відобразиться список усіх поточних замовлень, відсортований за номером столика, датою та статусом (рис. 5.3). Для створення нового замовлення, потрібно натиснути пункт «Добавить заказ», та заповнити необхідні поля, що з’являться на екрані (рисунк 5.4).

<div> ГЛАВНАЯ МЕНЮ ОТЧЕТ СОТРУДНИКИ ВЫХОД </div>							
<div> <h2>Заказы</h2> <p>Для редактирования конкретного заказа, дважды нажмите на него в списке ниже.</p> <p>Добавить заказ</p> </div>							
Код заказа	Номер столика	Дата оформления	Блюда	Стоимость (грн.)	Статус	Комментарий	
54	1	2020-06-06 17:14:33.878	Лимонно-базиликовый тарт (1 шт.), Салат с ростбифом и соусом Песто (1 шт.)	85.0	Новый	Без соли	✕
62	2	2020-06-06 17:17:01.411	Чизкейк из манго (1 шт.)	49.0	Новый		✕
65	3	2020-06-06 17:17:44.294	Цезарь с курицей (1 шт.), Эклер ванильный (3 шт.)	99.0	Новый		✕
87	3	2020-06-07 09:45:35.532	Чай зелёный с лимоном (2 шт.), Капучино (1 шт.)	32.0	Новый		✕
99	3	2020-10-22 13:28:47.568		0.0	Новый		✕
68	4	2020-06-06 17:18:49.248	Торт Наполеон (1 шт.), Фрукты ди Маре (3 шт.), Салат с ростбифом и соусом Песто (1 шт.)	293.0	Новый	Без острого	✕

Рис. 5.3 –сторінка замовлень

Введите данные заказа

Номер столика

3

Комментарий к заказу

Оформить заказ Отменить

Рис. 5.4 – спливаюче вікно створення замовлення

5.2.3 Зміна замовлення

Для зміни замовлення необхідно просто двічі натиснути на потрібному замовленні, та воно відкриється для редагування (рисунок 5.5).

Для видалення замовлення можна натиснути хрестик справа у списку замовлень навпроти потрібного замовлення (рисунок 5.3), або перейти до редагування замовлення та натиснути кнопку «Отменить заказ» там (рисунок 5.5).

Заказ №87

Столик - 3

Комментарий к заказу

Сохранить изменения

Блюдо	Количество	Стоимость (грн.)	
Чай зелёный с лимоном	2	14.0	✕
Капучино	1	18.0	✕

Добавить блюдо в заказ Отменить заказ

Рис. 5.5 – сторінка редагування замовлення

та вказати його кількість.

Змінити кількість замовленої страви можна двічі натиснувши на неї (відкриється сторінка редагування – рисунок 5.6). Видалити страву можна аналогічно замовленню – через меню редагування, або натиснувши хрестик навпроти потрібної страви.

Изменение заказанного блюда

Блюдо

Чай зелёный с лимоном

Количество

2

Применить изменения

Удалить блюдо из заказа

Рис. 5.6 – сторінка редагування замовленої страви

Для повара на сторінці редагування замовлення будуть доступні кнопки зміни статусу замовлення.

5.2.4 Взаємодія з меню

Будь-який відвідувач сайту може переглянути меню кафе. Для директора також доступні додаткові опції, а саме додавання, редагування та видалення страв з меню; та перегляд звітності щодо продажів. Операції роботи з меню повністю аналогічні операціям роботи з замовленнями (пункти 5.2.2-5.2.3).

Для перегляду звітності (рис. 5.7), директорові необхідно перейти на вкладку «Отчет».

Отчет по продажам		
Блюдо	Продано порций	Прибыль (грн.)
Эклер ванильный	7	161.0
Муссовый торт "Клубника - лайм - базилик"	4	196.0
Фрутти ди Маре	3	210.0
Цезарь с курицей	3	90.0
Чай зелёный с лимоном	2	14.0
Чизкейк "Сникерс"	2	118.0
Американо	2	40.0
Эспрессо	2	40.0
Салат с ростбифом и соусом Песто	2	80.0
Лимонно-базиликовый тарт	1	45.0
Торт Наполеон	1	43.0
Чизкейк из манго	1	49.0
Капучино	1	18.0
Чай с имбирём	0	0.0

Рис. 5.7 – сторінка звіту

5.3 Результати валідації програмного продукту

Внаслідок реалізації усіх запланованих компонентів, ПП вдалося вирішити поставлені перед ним цілі, а саме:

- Підвищено рівень доступності даних меню для клієнтів – меню та дані кафе доступні для перегляду онлайн.
- Підвищена швидкість обслуговування клієнтів - зменшився середній час оформлення замовлення клієнта – з 1 хвилини до 20 секунд.
- Трохи зменшився середній час вибору клієнтами закладу кафе страв для замовлень – з 4 хвилин до 3.
- Затрати часу на формування звітності знешкоджені – з 20 хвилин до «майже моментально»; дані популярності сраз стали більш доступними.
- Виключені помилки при математичних обчисленнях та необхідність повторних перевірок.

					ІКС КР 122 АІ181 ПЗ	Арк.
						38
Зм.	Арк.	№ докум.	Підпис	Дата		

Висновки до курсової роботи

В результаті створення програмного продукту була досягнута усі цілі.

Доказом цього є наступні факти:

- 1) зменшився середній час оформлення замовлення клієнта – з 1 хвилини до 20 секунд;
- 2) зменшився середній час вибору клієнтами закладу кафе страв для замовлень – з 4 хвилин до 3;
- 3) вдалося позбутися затрат часу на формування звітності – було 20 хвилин;
- 4) вдалося позбутися помилок при математичних обчисленнях та необхідності повторних перевірок.

В процесі створення програмного продукту виникли такі труднощі:

- 1) відсутність знань та досвіду в реалізації інтерактивних таблиць та модальних вікон;
- 2) велике навантаження з боку інших учбових дисциплін.

На щастя, вищеописані труднощі не вплинули негативно на реалізацію усіх прецедентів, але через обмежений час на створення програмного продукту, робота функцій могла бути недостатньо протестованою, або інтерфейс міг бути недостатньо інтуїтивним та зручним для користувача.

Зазначені недоробки планується виправити в майбутніх курсових роботах з урахуванням тем дисциплін наступних семестрів.

					ІКС КР 122 АІ181 ПЗ	Арк.
						39
Зм.	Арк.	№ докум.	Підпис	Дата		