

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій
Кафедра прикладних інформаційних систем**

122 «Комп'ютерні науки»
(шифр і назва спеціальності)

«Прикладне програмування»
(назва освітньої програми)

Кваліфікаційна робота бакалавра

на тему: «Веб-застосунок підтримки діяльності закладу громадського харчування»

Виконав _____
(Підпис)

Синельников Родіон Романович
(прізвище, ім'я, по батькові)

Керівник Домрачев Володимир Миколайович
(прізвище, ім'я, по батькові)

(Резолюція «До захисту»)

Попередній захист:

(Висновок: “До захисту в екзаменаційній комісії”)

Завідувач кафедри _____ Плескач В.Л.
(Дата) (Підпис) (Прізвище, ініціали)

Київ – 2021

**КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ
БАКАЛАВРА**

№ п/п	Назва етапів бакалаврської роботи	Термін виконання етапів бакалаврської роботи	Відмітка про виконання
1.	Вибір теми та наукового керівника бакалаврської роботи	26.10.2020	Виконано
2.	Видача завдання бакалаврської роботи	23.11.2020	Заява
3.	Настановча групова співбесіда з бакалаврської роботи	01.12.2020	Початок роботи з темою
4.	Затвердження плану бакалаврської роботи	18.02.2021	Виконано
5.	Підбір та вивчення літературних та інших джерел з теми дослідження	25.02.2021	Виконано
6.	Підготовка і подання науковому керівнику першого варіанту I розділу роботи	05.03.2021	Виконано
7.	Підготовка і подання науковому керівнику першого варіанту II розділу роботи	09.04.2021	Виконано
8.	Підготовка і подання науковому керівнику першого варіанту III розділу роботи	07.05.2021	Виконано
9.	Подання роботи у першому варіанті	11.05.2021	Виконано
10.	Оформлення пояснювальної записки бакалаврської роботи	12.05.2021	Виконано
11.	Подання бакалаврської роботи на попередній захист	24.05.2021	Виконано
12.	Врахування зауважень керівника і подання роботи в остаточному варіанті (з відповідним висновком про допуск) на кафедру	28.05.2021	Виконано
13.	Затвердження роботи в цілому (підготовка письмового відгуку керівника, письмова рецензія на бакалаврської роботу)	11.06.2021	Виконано
14.	Захист бакалаврської роботи	16.06.2021	Виконано

Здобувач вищої освіти _____

(підпис)

Керівник _____

(підпис)

ВІДОМІСТЬ ДИПЛОМНОЇ РОБОТИ

Зміст пояснювальної записки (перелік питань під час дослідження)

Складові частини дипломної роботи	Обсяг, арк. 54
Титульний аркуш	1 ст.
Календарний план виконання кваліфікаційної роботи бакалавра	1 ст.
Відомість дипломної роботи	1 ст.
Анотація	1 ст.
Анотація (іноземною мовою-англійською)	1 ст.
Зміст	2 ст.
Вступ	2 ст.
1. Загальносистемні питання. Аналіз розв'язуваної задачі й огляд наявних результатів. Постановка задачі на проектування	17 ст.
2. Проектні і технічні рішення. Види забезпечення	13 ст.
3. Реалізація і опис роботи системи	11 ст.
Висновки	2 ст.
Перелік посилань	2 ст.

				ДП ХХХХ 00.000.00		
	ПІБ	Підп.	Дата			
Розробн.				Відомість дипломної роботи	Лист	Листів
Керівн.						
Н/контр.	Макаренко С.А.					
Зав.каф.	Плескач В.Л.					

РЕФЕРАТ (АНОТАЦІЯ)

До бакалаврської дипломної роботи Синельникова Родіона Романовича на тему: «Веб-застосунок підтримки діяльності закладу громадського харчування»

Дана дипломна робота присвячена аналізу роботи закладу громадського харчування та розробці веб-застосунку для його покращення.

Об'єкт дослідження - веб-застосунок для підтримки роботи закладу громадського харчування, принципи та підходи організації програмної системи та функціональні рішення.

Предмет дослідження - програмні та технічні засоби для створення веб-застосунків, що були використані для розробки системи.

В роботі зроблено аналіз роботи закладу громадського харчування, вивчення особливостей діяльності працівників на певних посадах. Розроблено структуру локального серверу та бази даних.

Серверна частина розроблялася з використанням мови програмування Python та його фреймворку Django, клієнтська – використовуючи такі технології: HTML, CSS, JavaScript; база даних – MongoDB. Загальний об'єм роботи 54 сторінки, 31 рисунок, 3, 14 бібліографічних найменувань. Ключові слова: база даних, таблиця, локальний веб-сервер, веб-інтерфейс.

ANNOTATION (ABSTRACT)

To the diploma thesis by Synelnykov Rodion Romanovich
On the topic “Web application for supporting the activities of catering
establishments”

The aim of the diploma thesis is the analysis of the work of the catering establishment and the development of the web application to improve the work of the catering establishment.

The object of research is a web application to support the work of a catering establishment, the principles and approaches of organizing a software system and functional solutions.

The subject of research - software and hardware for creating web applications that were used to develop the system.

The work of the catering establishment, the study of the peculiarities of the activity of employees in certain positions is made in the work have been analyzed. The structure of the local web server and database have been developed.

Server part was developed using Python programming language and Django framework, client one – using HTML, CSS, JavaScript; database – MongoDB.

Total capacity: 54 pages, 31 figures, 14 references.

Tags: database, table, local web server, web interface.

ЗМІСТ

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА	2
ВІДОМІСТЬ ДИПЛОМНОЇ РОБОТИ.....	3
РЕФЕРАТ (АНОТАЦІЯ)	4
ANNOTATION (ABSTRACT)	5
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРИНЦИПУ РОБОТИ ЗАКЛАДУ ГРОМАДСЬКОГО ХАРЧУВАННЯ ТА ОБРАНИХ ДЛЯ РОЗРОБКИ ІНСТРУМЕНТІВ.....	10
1.1 Принцип роботи закладу громадського харчування	10
1.2 Аналіз обраних інструментів для розробки	20
1.2.1 Клієнт-серверна архітектура	20
1.2.2 Python як сучасна мова програмування.....	22
1.2.3 Django як інструмент створення сервера	23
1.2.4 MongoDB як сучасна нереляційна база даних.....	23
1.2.5 Мова гіпертекстової розмітки HTML.....	24
1.2.6 Каскадна таблиця стилів CSS.....	24
1.2.7 JavaScript як мова програмування інтерфейсу клієнтської частини клієнт-серверної архітектури	25
РОЗДІЛ 2. РОЗРОБКА ВЕБ-ЗАСТОСУНКУ ДЛЯ ПІДТРИМКИ РОБОТИ ЗАКЛАДУ ГРОМАДСЬКОГО ХАРЧУВАННЯ.....	27
2.1 Стратегія розробки веб-застосунку	27
2.1.1 Розробка клієнтської частини веб-застосунку	28
2.1.2 Створення бази даних	30
2.1.3 Сервер.....	36
2.2 Висновки	39
РОЗДІЛ 3. АНАЛІЗ ФУНКЦІОНАЛУ ВЕБ-ЗАСТОСУНКУ ТА ІНСТРУКЦІЯ КОРИСТУВАЧА	40
3.1 Основний функціонал веб-застосунку.....	40
3.1.1 Авторизація	41

3.1.2 Сервіс столів	42
3.1.3 Робота з замовленням	44
3.1.4 Офіс управління	47
3.1.5 Менеджмент приміщень	48
3.2 Висновок	49
ВИСНОВКИ	51
СПИСОК ЛІТЕРАТУРИ	53

ВСТУП

Система роботи закладу громадського харчування залежить від величезної кількості різних змінних. Для одиничного закладу важливим є місце, вид з вікон закладу, якість меблів та декору, не кажучи вже про основну його функцію – готувати замовлену гостем їжу та подавати її. Для ефективної організації роботи наймається чимала кількість професійних кадрів та використовуються найновіші технології управління, які ще потрібно «підігнати» під ті самі змінні. Для мережі таких закладів задача ускладнюється в десятки разів.

За останні роки було розроблено ефективні системи автоматизації роботи закладів громадського харчування. Вони дозволяють приділяти мінімум зусиль для комунікації між працівниками різних відділів, надаючи широкі можливості персоналу щодо організації передачі інформації та інструменти для зміни цієї самої системи під певні вимоги закладу. В своїй основі ця система спирається на три основні стовпи:

- Кіпери;
- Певна комп'ютерна програма, або локальний сервер;
- Відділ адміністраторів;

Кіпер – простий та невеликий за розміром комп'ютер, що підтримує можливості встановленої програми та дозволяє персоналу передавати інформацію між відділами за допомогою великих моніторів, що чутливі до дотику.

Комп'ютерна програма – серце і мозок системи. Ліцензія на певну комп'ютерну програму замовляється у компанії-розробника на певний термін з можливістю продовження дії договору. Вона встановлюється на центральний комп'ютер, до якого підключається мережа кіперів з усього закладу.

Відділ адміністраторів складається з одного, або декількох спеціалістів, що знайомі з системою, програмним та технічним обладнанням. Основна задача відділу – підтримка роботи системи та її адаптація під нові вимоги керівництва.

Розробка подібної системи коштує дорого, тож її використання можуть дозволити собі лише мережеві заклади громадського харчування та деякі заклади-одинаки. Саме для таких невеликих закладів було вирішено розробити прототип системи автоматизації роботи персоналу.

РОЗДІЛ 1. АНАЛІЗ ПРИНЦИПУ РОБОТИ ЗАКЛАДУ ГРОМАДСЬКОГО ХАРЧУВАННЯ ТА ОБРАНИХ ДЛЯ РОЗРОБКИ ІНСТРУМЕНТІВ

Для того, щоб створити систему підтримки діяльності закладу масового харчування потрібно виділити основні принципи його роботи в сучасному світі, виділити канали передачі інформації(комунікації) персоналу різного рівня та їх основні види взаємодії. Також потрібно визначитися з інструментами, за допомогою яких буде створено веб-застосунок. У цьому розділі згруповано основну інформацію щодо роботи закладу громадського харчування та способу створення системи його підтримки.

1.1 Принцип роботи закладу громадського харчування

Головна задача підприємства харчування – приготування їжі та подання її замовнику. Щоб приготувати їжу, подати її, помити посуд, прибрати стіл, тримати в чистоті всі приміщення, розраховувати гостей та керувати надходженням продуктів потрібен персонал. Для того, щоб цей персонал мав змогу виконувати свої обов'язки, їм потрібне місце, де будуть розміщені відповідні приміщення, цехи та зали для прийому гостей. Вся комунікація між персоналом повинна здійснюватись за допомогою зручних невеликих комп'ютерів – кіперів, що дозволить мінімізувати час та зусилля для передачі інформації між людьми.

Усі заклади громадського харчування повинні мати такі мінімальні набори груп приміщень:

- Складська;
- Виробнича;
- Адміністративна;
- Роздавальна.

Зклади громадського харчування повинні мати робочі місця для персоналу:

- Залу

- Офіціант;
- Бармен/бариста;
- Касир;
- Управління;
 - Менеджер по персоналу;
 - Менеджер по розвитку;
 - Менеджер по закупівлі;
- Кухні;
 - Шеф-кухар;
 - Кухар;
- Мийної;
 - Прибиральник;
 - Мийник посуду;

1.1.1 Приміщення закладу громадського харчування

1.1.1.1 Складські приміщення

До складу складських приміщень підприємств громадського харчування входять охолоджувальні камери для зберігання продуктів, що швидко псуються і напівфабрикатів, неохолоджувані комори для зберігання сухих продуктів, овочів.

Площа складського приміщення визначається в залежності від характеру роботи підприємства. Наприклад, в їдальнях, які працюють на сировині, охолоджувальні камери, а також комори для зберігання сухих продуктів повинні мати більшу площу, ніж на доготувальних підприємствах, де немає необхідності в зберіганні сировини. На невеликих підприємствах різні швидкопсувні продукти можуть зберігатися в одній камері, але в спеціально відведених для кожного виду продуктів місцях. На великих підприємствах м'ясні, рибні, гастрономічні продукти повинні зберігатися в окремих камерах. В ресторанах зазвичай окремі комори для хліба, а також приміщення для зберігання винних виробів, напоїв без вмісту алкоголю та фруктів.

Заготівельні підприємства, що постачають напівфабрикатами доготовочні, повинні мати в своєму складі експедиції з охолоджувальними камерами для короткочасного зберігання продукції перед відправкою.

При зберіганні різних груп товарів в складських приміщеннях підтримуються певний температурний режим і відповідна вологість. При розміщенні продуктів в коморах необхідно дотримуватись певних правил. Продукти не слід розміщувати поблизу водопровідних труб, опалювальної системи, охолоджувальних пристроїв. Від стін і підлоги продукти повинні бути розташовані на відстані не менше 20 см.

Приміщення для зберігання продуктів повинні відповідати певним санітарним вимогам. Стіни і стелі комор повинні бути гладкими і легко піддаватися санітарній обробці. У неохолоджуваних коморах допускається як природне, так і штучне освітлення, в коморах для зберігання овочів - тільки штучне освітлення.

1.1.1.2 Виробниче приміщення

До виробничих приміщень належать заготівельні, доготовчі цехи, а також мийна посуду та інвентарю закладу. Виробничі приміщення розташовують вздовж лінії технологічного процесу, щоб виключити зустрічні потоки сировини, що надходить, напівфабрикатів і готової продукції.

Виробничі приміщення розташовують на перших поверхах будівлі. Їх площа залежить від потужності підприємства. При влаштуванні виробничих приміщень необхідно забезпечувати зручне сполучення цехів зі складами та залом.

Гарячий цех - це цех, що являє собою кінцевий етап технологічного процесу приготування їжі, тут напівфабрикати перетворюються в готові страви. Він призначений для кулінарної обробки продуктів певними температурами. Гарячий цех повинен бути пов'язаний з усіма заготовочних цехами, холодним цехом, з залом через роздавальну і мийною кухонного посуду.

В приміщенні холодного цеху готуються закуски, холодні солодкі страви і напої, холодні супи. Також тут готові страви ділять на порції та оформляють. Холодний цех повинен бути пов'язаний з залом, мийною та гарячим цехом.

Холодний цех - це цех, в якому особливо строго дотримуються правила санітарії і правила особистої гігієни кухарів. Такий суворий підхід пояснюється тим, що продукція цеху після виготовлення, порціонування і оформлення не береться теплої обробці і повинна бути реалізована в короткий термін.

1.1.1.3 Адміністративне приміщення

Адміністративна група приміщень призначена для роботи керуючого персоналу. У ній знаходиться центральний комп'ютер системи а також зберігається крихкий інвентар (музичні інструменти). Це приміщення є центром комунікації в закладі громадського харчування. Також в адміністративному приміщенні проходять збори, наради управлінського персоналу, включаючи директорів та власника закладу.

1.1.1.4 Роздавальне приміщення

Роздавальна сполучає собою виробництво та зал. У ресторанах та кафе це приміщення напряму пов'язане з гарячим, холодним цехами та мийною столового посуду.

Роздавальні оснащуються прилавками для підтримки готового продукту в потрібному стані. Вони мають охолоджувальні прилавки для короткочасного зберігання холодних страв і закусок, солодких страв, холодних напоїв, кисломолочних продуктів, відповідні підставки для перших і других страв, прилавок з термостатом для заварювання чаю та створення гарячих напоїв, прилавки з касетами і підігрівом тарілок для перших та других страв, чашок для гарячих напоїв.

Персонал заходить з холу на територію роздавальної, де вільно переміщується між прилавками, вибирає відповідні страви, завдяки чому йому

не доводиться безпосередньо взаємодіяти з цехами на кухні. З стравами на розносі він залишають роздавальну дуже скоро. Таким чином, роздавальна вільного руху дозволяє персоналу швидше та з певним комфортом підтримувати потрібну для закладу якість обслуговування та сервісу.

1.1.2 Персонал закладу громадського харчування

1.1.2.1 Офіціант

Основне завдання офіціанта -обслуговування відвідувачів закладу громадського харчування. Кожен заклад висуває індивідуальні вимоги до роботи персоналу обслуговування. Згідно з вимогами, офіціанти зустрічають клієнтів, займаються сервіруванням столу та подачею страв, відповідно до заданого за зразком і загальноприйнятій етикету. Крім цього вони зобов'язані носити певну уніформу і бейдж з посадою та ім'ям. Наприклад, існують специфічні ресторани, де панує темрява, тому до складу уніформи офіціанта включено прилади нічного бачення.

До основних вимог закладу є знання офіціантом меню свого закладу, уміння розповісти про технологію приготування страв, знати їх характеристику і ціни. Також він повинен надавати рекомендації клієнтам щодо страв, що підходять по смаку, та напоїв, що гарно смакують з певними стравами.

Основний обов'язок офіціанта - це прийом замовлень від відвідувачів. Він повинен чітко та зрозуміло відповісти на будь-які питання гостей з приводу меню. Офіціант записує або запам'ятовує замовлення, при необхідності уточнюючи його. Після чого він повинен оперативно і точно передати його кухареві і бармену. Офіціант контролює своєчасність приготування і зовнішній вигляд страв і подає їх гостям.

1.1.2.2 Бармен

Основне завдання бармена полягає в приготуванні, розливі та оформленні напоїв. Однак часто так буває, що саме бармен стає ключовою

фігурою того чи іншого закладу громадського харчування, будь то бар, кафе, ресторан або клуб.

Бармен приймає замовлення від відвідувачів, консультує відвідувачів з питань замовлення страв та напоїв, обслуговує відвідувачів за барною стійкою готуваними до вживання безалкогольними і алкогольними напоями та стравами, готує безалкогольні та алкогольні коктейлі, оформляє вітрини і барну стійку, містить їх в порядку, експлуатує і обслуговує музичну апаратуру.

1.1.2.3 Касир

Основне завдання касира зводиться до розрахунку, видачі і прийому фінансових коштів. Крім цього, він оформляє документацію і видає фіскальні чеки. Відповідно, він повинен вміти поводитися з касовим апаратом. Робоче місце - адміністративне приміщення ресторану, або кафе, або прилавок, в закладі громадського харчування швидкого обслуговування.

Здійснювати операції по прийому, обліку, видачі і зберіганню грошових коштів з видачею чека покупцеві, використовуючи в роботі контрольно касову машину.

Касир знає асортимент продукції кафе за назвою і зовнішнім виглядом, передає відповідно до встановленого порядку грошові кошти в бухгалтерію закладу, становить касову звітність, дотримується вимог пожежної безпеки, містить в чистоті своє робоче місце, виконує правила експлуатації обчислювальної техніки і контрольно- касової машини, дотримується затверджену адміністрацією форму одягу і зовнішній вигляд.

1.1.2.4 Менеджер

Менеджер ресторану займається організацією взаємодії всіх структурних підрозділів ресторану, організацією праці персоналу, забезпеченням і контролем наявності необхідних запасів, контролем технічного стану технологічного обладнання; веденням документації; дотриманням правил охорони праці, техніки безпеки, пожежної безпеки.

Менеджер ресторану несе відповідальність за поганий стан і неякісне приготування страв, а також за будь-які скарги клієнта, за невиконання службових обов'язків, за нанесення матеріальних збитків.

Основний обов'язок менеджера ресторану - це забезпечення якісного та ефективного обслуговування відвідувачів. Виходячи з цього, менеджер ресторану виконує наступні обов'язки:

- організація і проведення банкетів;
- розподіл обов'язків між співробітниками ресторану;
- навчання, адаптація, мотивація персоналу;
- контроль за дотриманням санітарних норм і правил;
- контроль за дотриманням стандартів обслуговування і процесів приготування страв;
- контроль за дотриманням правил і норм техніки безпеки та охорони праці;
- контроль термінів придатності продуктів, проведення інвентаризацій;
- ведення касової звітності, інкасація грошових коштів;
- запобігання і вирішення спірних і конфліктних ситуацій за участю відвідувачів і співробітників ресторану.

1.1.2.5 Кухар

Посадові обов'язки кухаря залежать від величини і профілю закладу громадського харчування. Він виконує посадові обов'язки, такі як: безпосередньо здійснювати приготування страв, в тому числі: мийку продуктів, змішування продуктів, смаження, випікання, варіння на пару, приготування соусів, супів, бульйонів, холодних закусок для шведського столу і салатів, декорування страви, планування меню, виходячи з можливостей закладу і своїх власних, вивчення вимог клієнтів до обслуговування і якості страв і продуктів, проведення інструктажів офіціантам, контроль роботи по прибиранню, дезінфекції, санітарної обробки

службових та виробничих приміщень, вивчення скарг і претензій гостей до якості страв і обслуговування, ведення статистичного обліку скарг і претензій, приготування пропозицій щодо вдосконалення роботи.

1.1.2.6 Прибиральник

Прибиральник закладу громадського харчування – член персоналу, який повинен підтримувати чистоту усіх приміщень закладу, підтримувати у готовності до роботи робочі інструменти та швидко реагувати на запит з певного приміщення.

1.1.3 Комунікація між персоналом

Для того, щоб заклад громадського харчування працював ефективно, необхідна постійна активна комунікація між персоналом в різних приміщеннях. Раніше інформація передавалася за допомогою записок, чи навіть самотужки людина повинна була прийти в інший відділ. Зараз же переважна більшість закладів громадського харчування, в особливості ресторани та кафе, виділяють частину бюджету для закупівлі технічної апаратури, оренді та продовженню співпраці з постачальником програмного забезпечення, лише для того, щоб мінімізувати час та зусилля на комунікацію.

Після спостереження за роботою персоналу ресторану, було проаналізовано комунікацію персоналу. Результатом спостереження та аналізу є висновок, що в кожному закладі громадського харчування є два цикли комунікації: зовнішній та внутрішній.

1.1.3.1 Внутрішній цикл комунікації

Внутрішній цикл складається з трьох основних фаз, а саме: початок, власне, сам цикл та вихід з нього. Початком є відкриття закладу: працівники кухні інформують адміністрацію та сервісний персонал про стан кухні: відсутність, чи нестача певних інгредієнтів, страви, які потрібно продати

якнайшвидше; сервісний персонал інформує адміністрацію про готовність приміщень приймати гостей.

Власне, сам цикл розпочинається з приходом першого гостя. Відвідувач інформує своєю присутністю сервісний персонал та подає знак, що готовий зробити замовлення. Після прийняття замовлення, офіціант повинен занести дані до системи за допомогою кіпера. Далі кухар на своєму пристрої отримує повідомлення та друковану пояснювальну записку що та звідки замовили, час, коли потрібно приготувати та певні коментарі від сервісного персоналу. Кухар відмічає на своєму кіпері, що повідомлення отримано, та виконує свою роботу. По завершенню роботи він, або помічник кухаря, виносить страву на роздавальну разом з друкованою пояснювальною запискою, та відмічає на своєму кіпері, що страва готова. Після цього в орієнтований час приготування, офіціант виходить з залу та знаходить за допомогою пояснювальної записки свою страву та подає її замовнику. Ці дії можуть бути повторні за бажанням гостя. Після того, як відвідувач вирішив отримати рахунок, офіціант за допомогою кіпера роздруковує чек та повідомляє касира про потребу в розрахуванні.

Завершення циклу настає після закриття та розрахунку останнього гостя. Кухня повідомляє адміністрацію про частоту замовлень певних страв та нестачу певних продуктів, які повинна замовити адміністрація та врахувати потенціальний зріст попиту на певні страви. Сервісний персонал готує зали для прийому гостей до наступного дня та повідомляє про це адміністрацію.

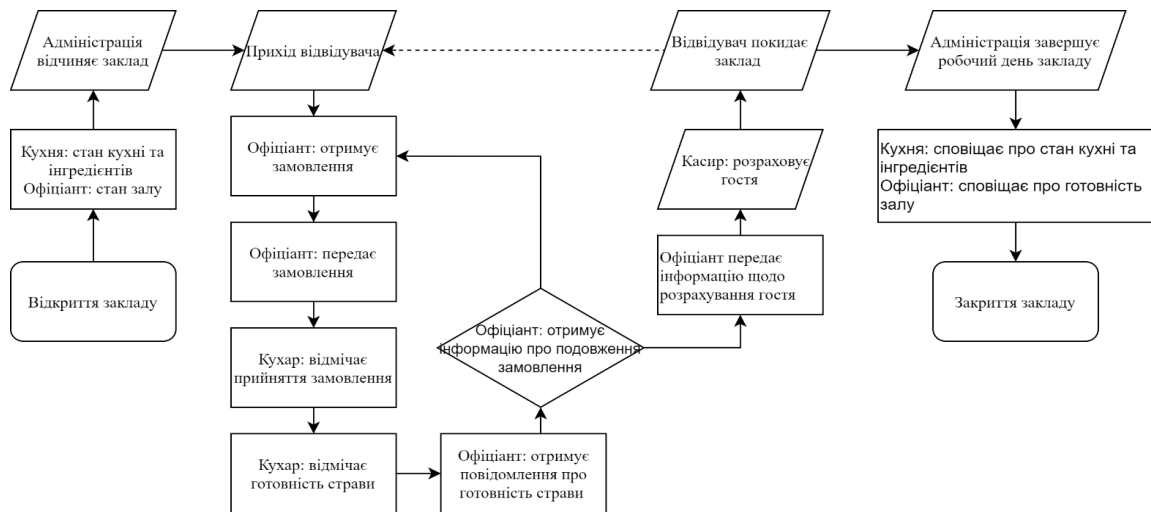


Рисунок 1.1 Внутрішній цикл комунікації

Після завершення внутрішнього циклу комунікації, заклад громадського харчування зачиняють, щоб розпочати цей цикл спочатку наступного дня.

1.1.3.2 Зовнішній цикл комунікації

Зовнішній цикл комунікації розпочинається з закінченням робочого дня. Адміністрація підраховує потенційний попит на певні страви та аналізує інформацію, надану кухарами з різних цехів та формує запит на доставку інгредієнтів від постачальника та корегують меню закладу (розширити вибір певних категорій страв, що користуються більшим попитом та зменшити надходження інгредієнтів до менш цікавих споживачам страв, або навіть прибрати їх з меню). Для затвердження правок в меню, адміністрація консультується з кухарами, щодо можливостей розширення асортименту.

Перед відкриттям закладу, постачальник доставляє замовлені продукти в повному, або частковому обсязі. У випадку неповної доставки, адміністрація інформує кухарський персонал щодо затримок та роблять запит на обмежене використання певних інгредієнтів шляхом тимчасового викреслювання страви з меню. Після отримання звіту щодо тимчасової відсутності певних страв, адміністрація повідомляє сервісний персонал щодо їх неможливості замовлення.

1.2 Аналіз обраних інструментів для розробки

Для розробки веб-застосунку для підтримки діяльності роботи закладу громадського харчування було вирішено створити клієнт-серверну архітектуру з локальним веб-сервером, який знаходиться на головному комп'ютері закладу, та до якого підключено робочі пристрої(кіпери) для персоналу. Для створення сервера було обрано мову програмування Python, а саме його фреймворк для створення сучасного сервера Django. Для зберігання даних було обрано нереляційну базу даних MongoDB через те, що вона добре підходить для запису даних про замовлення. Для створення клієнтської частини архітектури було обрано мову гіпертекстової розмітки HTML, каскадну таблицю стилів CSS та мультипарадигмальну мову програмування JavaScript.

1.2.1 Клієнт-серверна архітектура

Комп'ютери і програми, що входять до складу інформаційної системи, не є рівноправними. Деякі з них володіють ресурсами (файлова система, процесор, принтер, база даних і т. д.), інші мають можливість звертатися до цих ресурсів. Комп'ютер (або програму), керуючий ресурсом, називають сервером цього ресурсу (файл-сервер, сервер бази даних, обчислювальний сервер).

Клієнт і сервер будь-якого ресурсу можуть знаходитися як на одному комп'ютері, так і на різних комп'ютерах, пов'язаних мережею. Архітектура "клієнт-сервер» визначає загальні принципи організації взаємодії в мережі, де є сервери, вузли-постачальники деяких специфічних функцій (сервісів) і клієнти, споживачі цих функцій.

Двофазна архітектура використовується в клієнт-серверних системах, де сервер відповідає на клієнтські запити безпосередньо і в повному обсязі, при цьому використовуючи тільки власні ресурси. Тобто сервер не викликає сторонні мережеві програми, але не звертається до сторонніх ресурсів для виконання будь-якої частини запиту (рис.1.2)

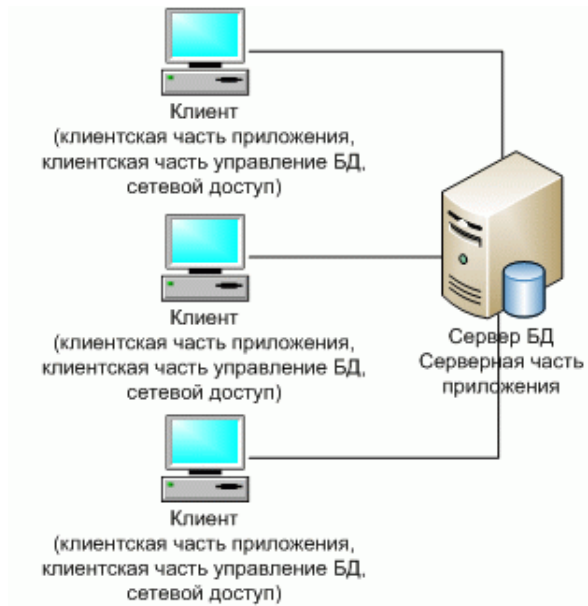


Рисунок 1.2 Двофазна клієнт-серверна архітектура

Ще одна тенденція в клієнт-серверних технологіях пов'язана з дедалі більшим використанням розподілених обчислень. Вони реалізуються на основі моделі сервера додатків, де мережевий додаток розділене на дві і більше частин, кожна з яких може виконуватися на окремому комп'ютері. Виділені частини додатка взаємодіють один з одним, обмінюючись повідомленнями в заздалегідь узгодженому форматі. В цьому випадку двофазна клієнт-серверна архітектура стає трифазною (рис.1.3).

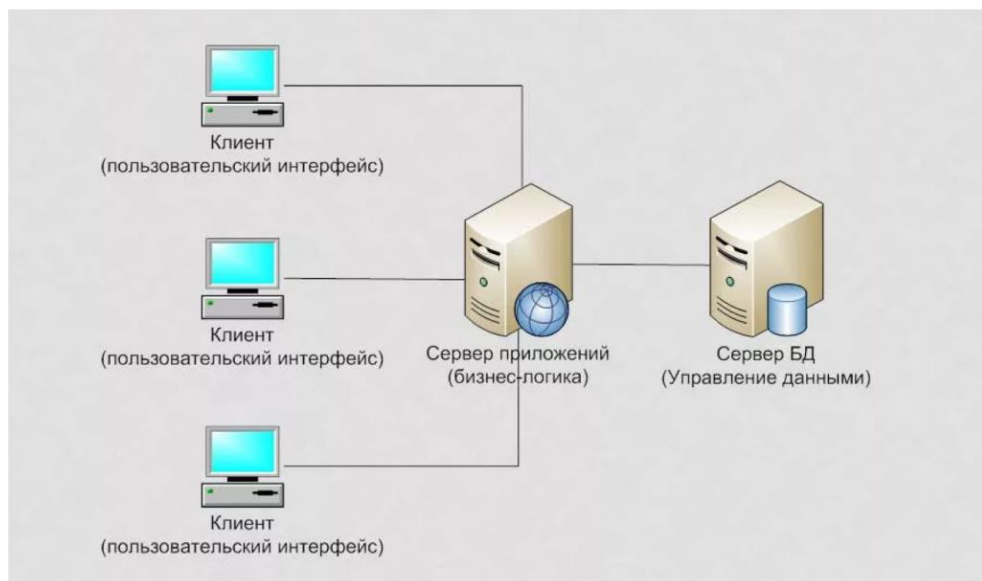


Рисунок 1.3 Трифазна клієнт-серверна архітектура

Як правило, третьою ланкою в триланковій архітектурі(three-tier, 3-tier) стає сервер додатків.

1.2.2 Python як сучасна мова програмування

Python – це мова скриптів, що активно розвивається та яка використовується для вирішення самих різнопланових проблем і завдань. Вона знадобиться в створенні додатків на усі платформи, Її застосовують в роботі з великим об'ємом інформації, при розробці веб-сайтів та використовують в машинному навчанні. Дану мову програмування використовують великі відомі корпорації.

Основні можливості Python:

- Об'єктно-орієнтоване програмування (ООП).
- Узагальнене і функціональне програмування.
- Модулі та пакети. Програмне забезпечення на Python оформлюється у вигляді модулів, які можуть бути зібрані в пакети.
- Інтроекція. Це можливість запросити тип і структуру об'єкту під час виконання програми. У Пітоні для будь-якого об'єкта можна дізнатися всю інформацію про його внутрішню структуру.
- Обробка винятків, ітератори і генератори.
- Управління контекстом виконання.
- Декоратори, регулярні вирази

Переваги цієї мови:

- Мова характеризується логічним синтаксисом
- Код програм легко читається і сприймається.
- Велика інтернет-спільнота, що відповіла на більшість можливих питань, пов'язаних з цією мовою.
- Гнучкість і масштабованість.
- Python - інтерпретована мова програмування, що дозволяє розробникам працювати з нею на всіх платформах.

1.2.3 Django як інструмент створення сервера

Django - безкоштовний фреймворк для веб-додатків, написаний на Python. Кожен раз при розробці веб-сайтів потрібні схожі компоненти: спосіб автентифікувати користувачів (вхід, вихід, реєстрація), панель управління сайтом, форми, інструменти для завантаження файлів.

Коли на сервер приходить запит, він переадресовується Django, який розбирається, що від нього просять. Для початку він бере адреса веб-сторінки і використовує `urlresolver`, бере список шаблонів і намагається зіставити їх з URL. Django звіряє шаблони зверху вниз і, якщо щось збігається, він перенаправляє запит відповідної функції (яка називається `view`).

Функція `view` може перевірити, чи має користувач дозвіл зробити запит до бази даних, а потім оновить опис роботи і відправить назад відповідь. Потім функція `view` генерує відповідь, і Django зможе відправити його веб-браузеру користувача.

1.2.4 MongoDB як сучасна нереляційна база даних

MongoDB - це документно-орієнтована база даних NoSQL, використовувана для зберігання великих обсягів даних. Замість використання таблиць і рядків, як в традиційних реляційних базах даних, MongoDB використовує колекції і документи. Документи складаються з пар ключ-значення, які є основною одиницею даних в MongoDB. Колекції містять набори документів і функції, які еквівалентні таблицями реляційної бази даних. MongoDB - це база даних, яка з'явилася в середині 2000-х років.

Кожна база даних містить колекції, які в свою чергу містять документи. Кожен документ може бути різним з різною кількістю полів. Розмір і зміст кожного документа можуть відрізнятися один від одного.

Структура документа більше відповідає тому, як розробники конструюють свої класи і об'єкти на відповідних мовах програмування. Розробники часто говорять, що їх класи не є рядками і стовпцями, а мають чітку структуру з парами ключ-значення. Рядки (або документи, що

викликаються в MongoDB) не обов'язково повинні мати наперед визначену схему. Замість цього поля можуть бути створені на льоту.

Модель даних, доступна в MongoDB, дозволяє вам представляти ієрархічні відносини, простіше зберігати масиви та інші більш складні структури.

Середовища MongoDB дуже масштабуються. Компанії по всьому світу визначили кластери, причому деякі з них мають більше 100 вузлів з мільйонами документів в базі даних.

1.2.5 Мова гіпертекстової розмітки HTML

HTML (Hypertext Markup Language) - це код, який використовується для структурування і відображення веб-сторінки і її контенту. Наприклад, контент може бути структурований всередині безлічі параграфів, маркованих списків або з використанням зображень і таблиць даних.

HTML не є мовою програмування; це мова розмітки, і використовується, щоб повідомляти вашому браузеру, як відображати веб-сторінки, які ви відвідуєте. Він може бути складним або простим, в залежності від того, як хоче веб-дизайнер. HTML складається з ряду елементів, які використовуються, щоб вкладати або обертати різні частини контенту, щоб змусити контент відображатися або діяти певним чином. Огороджувальні теги можуть зробити слово або зображення посиланням на щось ще, можуть зробити слова курсивом, зробити шрифт більше або менше і так далі.

1.2.6 Каскадна таблиця стилів CSS

CSS - це мова, за допомогою якого описується зовнішній вигляд документа HTML, XML, XHTML. Назва означає «каскадна таблиця стилів», або Cascading Style Sheets. CSS-стилі незамінні при оформленні сторінок сайтів: в одному файлі містяться відомості про відображення всіх елементів документа.

По суті, таблиця стилів - це файл, де описується, як буде виглядати кожен з елементів на сторінці. У HTML-документі, таким чином, залишається

тільки структура сторінки: самі блоки, їх вміст і розташування. Створити сторінку і оформити її можна і без використання таблиць, прописуючи візуальні властивості кожного елемента в його описі. Але, якщо сторінок сотні і тисячі, застосовувати такий метод незручно: при зміні оформлення доводиться міняти безліч документів, до того ж це ускладнює написання коду. Тому використання CSS вважається золотим стандартом оформлення сайтів: так вийшло завдяки гнучкості і різноманіттю можливостей каскадних таблиць.

1.2.7 JavaScript як мова програмування інтерфейсу клієнтської частини клієнт-серверної архітектури

JavaScript - це мова програмування, який додає інтерактивність на веб-сайт (наприклад: ігри, відгук при натисканні кнопок або при введенні даних у форми, динамічні стилі, анімація).

JavaScript - це повноцінний динамічний мову програмування, який застосовується до HTML документу, і може забезпечити динамічну інтерактивність на веб-сайтах.

JavaScript неймовірно універсальний. Маючи великий досвід, можливо створювати ігри, анімовану 2D і 3D графіку, повномасштабні програми з базами даних і так далі.

JavaScript сам по собі досить компактний, але дуже гнучкий. Розробниками написано велику кількість інструментів поверх основного мови JavaScript, які розблокують величезна кількість додаткових функцій з дуже невеликим зусиллям. До них відносяться:

- Програмні інтерфейси додатка (API), вбудовані в браузері, що забезпечують різні функціональні можливості, такі як динамічне створення HTML і установку CSS стилів, захоплення і маніпуляція відеопотоком, робота з веб-камерою користувача або генерація 3D графіки.

- Сторонні АРІ дозволяють розробникам впроваджувати функціональність в свої сайти від інших розробників, таких як Twitter або Facebook.
- Застосування до HTML сторонніх фреймворків і бібліотек, що дозволяють прискорити створення сайтів і додатків.

1.3 Результат проведеного аналізу

Після проведення аналізу роботи закладу громадського харчування було сформовано дві схеми потоків інформації та комунікацій між персоналом різних відділів та приміщень, було виявлено необхідні складові кожного закладу громадського харчування та розроблено схему швидкої комунікації між ними. Для створення веб-додатку було обрано певний пакет програм та мов програмування, що добре підходять для його реалізації.

РОЗДІЛ 2. РОЗРОБКА ВЕБ-ЗАСТОСУНКУ ДЛЯ ПІДТРИМКИ РОБОТИ ЗАКЛАДУ ГРОМАДСЬКОГО ХАРЧУВАННЯ

В даному розділі описується розробка веб-застосунку, її окремі етапи та загальний підхід до створення клієнт-серверної архітектури. Насамперед потрібно розібрати стратегію розробки веб-сервісу, потім розібрати її поетапно.

2.1 Стратегія розробки веб-застосунку

Оскільки кінцевий споживач розробленого продукту - працівники закладу громадського харчування, стратегія розробки спирається на надання максимально комфортного використання веб-застосунку. Для демонстрації стратегії розробки було створено схему(рис. 2.1), що відображає основні етапи створення веб-застосунку.

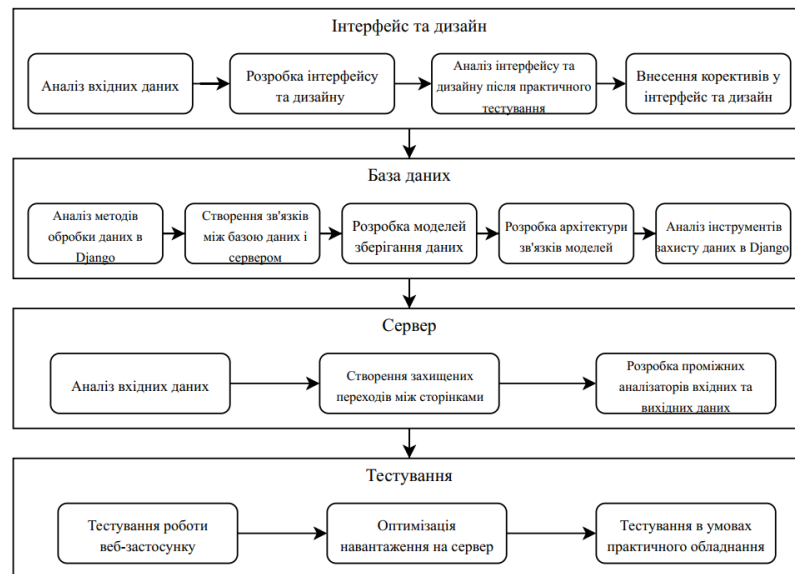


Рисунок 2.1 Схема етапів створення веб-застосунку

Веб застосунок створювався на основі клієнт-серверної архітектури та має такі основні етапи реалізації:

- Розробка клієнтської частини застосунку
- Створення бази даних
- Розробка алгоритмів роботи сервера
- Тестування в умовах роботи

2.1.1 Розробка клієнтської частини веб-застосунку

Дизайн та можливості інтерфейсу були розроблені з мінімальною кількістю елементів, що привертають увагу працівника. Основним завданням розробки дизайну та інтерфейсу було створення простих у розумінні та роботі об'єктів. Так як цільовий користувач – людина, що майже ніколи не матиме можливості взаємодіяти з веб-застосунком за допомогою мишки, було розроблено специфічний дизайн, що мав невелику кількість інтерактивних елементів, проте чудово взаємодіяв з екранами, чутливими до дотику – кіперами.

Першим етапом розробки клієнтської частини застосунку є аналіз вхідних даних. Кожна сторінка потребувала свій особливий набір даних, отже було вирішено для створення робочого інтерфейсу завантажувати попередньо створені статичні дані. Переважна більшість сторінок потребувала для коректної роботи дані про користувача, його посаду та особливий номер. Крім того, сторінки, що працюють із замовленнями, потребували, наприклад, переліку всіх страв, що є у базі, страв, що вже попередньо були замовлені, дані про столи, на які можна зробити замовлення та приміщення, де знаходиться той самий стіл. Після створення бази даних, статичні дані було замінено результатами запитів до бази даних.

Дизайн, для запобігання помилок, був створений з мінімальними можливостями власноручно змінити значення полів(рис. 2.2).

Васк <	Паста	Супи
Салати	Основні	Закуси
Піца	Десерти	
Тірамісу		

Офіціант: Зубенко Михайло Петрович		Стіл №: 1		Гостей: 2	
Замовлення № 2021051822261811		Зал: Літня тераса		Позиції: 3	
Назва	Кількість	На винос	Колі готувати	Ціна	
Стейк з телятини	- 2 +	<input checked="" type="checkbox"/>	Чекати 5 хвилин	520	X
Неаполітана	- 3 +	<input type="checkbox"/>	Не чекати	450	X
Тірамісу	- 1 +	<input checked="" type="checkbox"/>	Чекати 20 хвилин	105	X
Сума				1075	
Додати		ЧЕК		X	

Рисунок 2.2 Інтерфейс персоналу обслуговування

Можливість самостійно редагувати дані є лише у керівного персоналу, такого як менеджери, директор, чи власник, або адміністратор(рис.2.3). Ця можливість стає доступною лише при використанні головного комп'ютера закладу.

Введіть назву нового залу	
<input type="text" value="КНУ_ФІТ"/>	
Відмінити	Продовжити

Рисунок 2.3 Інтерфейс з можливістю власноручно ввести дані

Для проведення тестування інтерфейсу та дизайну веб-застосунку сторінки було запущено на сенсорному екрані, що виявило ряд проблем, які уповільнюють процес роботи, або зовсім його зупиняють. Дизайн було перероблено з урахуванням екранів будь-якої роздільної здатності. Також

було вирішено проблему коректного відображення вже створених замовлень. Після повторного тестування, помилок та зауважень виявлено не було.

2.1.2 Створення бази даних

Наступним етапом реалізації клієнт-серверної архітектури веб-застосунку є створення бази даних. Django не дозволяє серверу напряму отримувати дані з бази, тому пропускає потоки даних у вигляді запитів через певні алгоритми обробки. За допомогою цих алгоритмів не потрібно писати запити до бази даних власноруч. Потрібно лише вказати таблицю, з якої потрібно отримати дані. Для того, щоб фільтрувати дані Django надає можливість фільтрувати запит. У цьому фільтрі потрібно лише вказати яке поле, чи поля потрібно порівняти, та іншим параметром є змінна, з якою потрібно зрівняти. Важливо зазначити, що поки та змінна, в яку ми заносимо інструкції для створення запиту, не буде явно надіслана клієнту, вона не зробить запит до бази даних. Це означає, що за допомогою одного глобального запиту можна створити чималу кількість складних запитів, пишучи мінімальну кількість коду. Але у таких запитів буде вагомий мінус. Кожен такий запит є унікальним, тому він окремо від інших запитів взаємодітиме з базою даних.

Для підключення до бази даних, достатньо вже в попередньо створених меню настройки, вказати СУБД та саму базу даних, з якою буде взаємодія (також іноді потрібно вказати логін та пароль)(рис.2.4).

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

Рисунок 2.4 Приклад встановлення з'єднання з базою даних

Для створення таблиці в Django також не потрібно взаємодіяти з СУБД, чи базою даних напряму. Окремий додаток настройки таблиць бази даних

вбудований в Django. Кожна таблиця являє собою модель, яку Django інтерпретує як інструкцію по зберіганню, зміні та видаленню даних. Модель представляє собою клас, до об'єктів якого і звертаються при створенні запиту до бази даних(рис. 2.5).

```
try:
    order_id_check = Orders.objects.get(order_number=order_number)
```

Рисунок 2.5 Приклад запиту за допомогою об'єктів класу(Orders)

Опис полів та методів класу Django дозволяє вести у відповідному файлі. Мінімальний опис має складати хоча б одне поле (крім поля індивідуального номера. Django додає його самостійно) та спосіб відображення об'єктів цього класу у вигляді записів у базі даних (рис. 2.6). Також можливо додати ключові слова, за допомогою яких адміністратор зможе знайти цю таблицю.

```
class Dishes(models.Model):
    name = models.CharField('Назва', max_length=50)
    description = models.TextField('Опис страви')
    price = models.IntegerField('Ціна', default='1')
    ingredients_consists = models.TextField('Які інгредієнти містить', default='')
    foodlink = models.CharField('Назва категорії, до якої належить страву', max_length=50)
    name_of_class = models.CharField('Назва класу при вигразці', max_length=50, default='')

    def __str__(self):
        return self.name

    class Meta:
        verbose_name = 'Страва'
        verbose_name_plural = 'Страви'
```

Рисунок 2.6 Приклад створення таблиці в Django

Django надає широкі можливості роботи з таблицями з фіксованою кількістю полів, отже й сама архітектура бази даних веб-застосунку розроблена на основі роботи СУБД SQLite. Схема бази даних зображена на рисунку 2.7.



Рисунок 2.7 Схема таблиц бази даних та зв'язків між ними

Для повного розуміння роботи бази даних, потрібно проаналізувати основні таблиці, з якими відбувається більша частина роботи, зрозуміти як взаємодіють таблиці між собою та зі сторінками, з якими обмінюються інформацією за допомогою сервера.

2.1.2.1 Робітники

Таблиця робітників – основна таблиця, дані з якої використовуються при кожній взаємодії з базою даних задля забезпечення безпеки роботи веб-застосунку. В таблиці працівників вказаний їхній особистий пароль, за допомогою якого вони авторизуються в системі, оформлюють замовлення на своє ім'я, розпочинають, або завершують свій робочий день (дані про початок та кінець робочого дня заносить в систему сам працівник за допомогою першої авторизації в системі на початку робочого дня, та натисканням спеціальної кнопки для завершення робочого дня. В системі зберігається точний час, що є різницею між початком та кінцем зміни. Докладніше у розділі 3).

Таблиця посад зберігає дані про види посад, що є в системі. Ієрархія посад реалізується наданням персоналу різного рівня доступу, що забезпечує диференціацію між персоналом різного рівня доступу, або тих, хто працює в різних відділах. Так, персонал, основною задачею якого є обслуговування, не може отримати доступ до сторінок керування персоналом, доступ до якого має лише керівний персонал, або до даних про платню, чи кількість відпрацьованого часу, доступ до яких має лише власник закладу. Важливо зазначити, що керівний персонал, власник закладу, або адміністратор можуть власноруч змінювати деякі дані в базі даних, але Django веде повний аудит роботи з базою даних, записуючи кожен дію, що була зроблена, тому завжди можна побачити, хто, та до яких даних хотів отримати доступ, та які поля було змінено в базі даних.

2.1.2.2 Столи та приміщення

Столи є основною одиницею для прив'язки нового відкритого замовлення. За допомогою столів, персонал, завдання якого є сервіс, виконує перший крок в роботі системи – вносить дані. Столи мають свій унікальний номер в базі даних, але в системі майже завжди користувач використовує номер столу, що є унікальним для певного приміщення, але може повторюватися в різних приміщеннях. Дані про столи, що допомагають їм коректно відображатися в системі – це їх координати. Система відображення столів працює за допомогою невидимої сітки, в комірки якої за певними координатами, сервер завантажує дані, які відображаються як стіл. Важливо зазначити, що це не остаточне вирішення проблеми відображення столів. Система розрахована на будь-яку кількість приміщень, але на цій стадії, програма може умістити в фіксовану кількість комірок невидимої сітки лише 50 столів. Директор та власник можуть власноруч за допомогою відповідного інтерфейсу вносити зміни в координати столів в певних приміщеннях, або навіть створити нове. Сторінка роботи зі столами та залами не є доступною під

час роботи закладу. Лише після завершення роботи, коли всі столи будуть без замовлень, з'являється можливість працювати з цим інтерфейсом.

Приміщення(зали в системі) – найбільша одиниця для прив'язки даних – столів. На даному етапі розробки програми, зали мають можливість уміщувати до 50 столів(сітка розміром 5 на 10). У майбутньому буде надана можливість самостійно розміщувати столи, та не буде ліміту для кількості столів в одному приміщенні. Це стане можливим з використанням технології Drag-and-Drop, що дозволить користувачам з відповідним рівнем доступу не спиратися на попередньо задані координати.

2.1.2.3 Страви та інгредієнти

В базі даних страви представлені трьома взаємопов'язаними таблицями з ієрархічною структурою. Кожна страва відноситься до певної категорії, наприклад: супи, чи десерти. Кожна категорія відноситься до виду, яких, зазвичай, в закладах громадського харчування три: страви, напої та інше. Кожна страва містить опис та список інгредієнтів. Це потрібно для того, щоб дозволити персоналу вказати в замовленні, що певна страва потребує вилучення з неї одного, або декількох інгредієнтів. Така ситуація виникає не часто, але завжди в заклад може завітати людина з алергією на певні продукти. Ця можливість дає змогу інформувати кухара про особливості замовлення без зайвої потреби комунікувати з ним особисто. Опис страви та список інгредієнтів також допомагають персоналу на стадії стажування на практиці вивчати потрібний матеріал та особливості певних страв.

Таблиця інгредієнтів також складається з двох підтаблиць. Перша – це, власне, список інгредієнтів. У кожного з них є прив'язка до певної категорії інгредієнтів(такі як м'ясо, риба, зелень, молочні продукти, м'ясо). Ці категорії перераховані в другій підтаблиці. Дана структура зроблена для мінімізації часу при пошуку потрібних інгредієнтів, при оформленні замовлення. Особливості страви вказуються в певному меню, в якому потрібно вибрати категорію, а потім знайти потрібний продукт, який гість побажав додати до своєї страви

(планується реалізувати в веб-застосунку в найближчий час). В сучасному світі заклади громадського харчування пильно слідкують за тим, щоб у їхньому меню були страви з мінімальною кількістю алергенів, але зовсім без них створити меню неможливо. Саме тому кожен інгредієнт, якщо він потенційно є алергеном, помічається в базі даних відповідним полем, що дозволяє персоналу швидко надати рекомендацію щодо страв, які не мають у своєму рецепті певних алергенів, адже знання алергенів необхідно для усіх працівників закладу.

2.1.2.4 Заовлення та позиції

Оформлення заовлення – перший крок в процесі обміну інформацією між персоналом в системі. Для того, щоб таблиця заовлень не була перевантажена різною кількістю страв, що включає в себе заовлення та для того, щоб уникнути труднощів при роботі з такими даними було вирішено розбити таблицю заовлень на дві окремі, пов'язані між собою таблиці: власне, список заовлень, та таблиця позицій.

Таблиця заовлень прив'язана до таблиці столів, бо на даний момент на один стіл може припадати лише одне заовлення(планується додати можливість створити більше одного заовлення на один стіл), тож ця таблиця зберігає дані столу, на який було оформлене заовлення, унікальний код, що генерує сама база даних, та свій код заовлення, що генерується за певним алгоритмом та також є унікальним. Саме ей код заовлення буде відображатися на столі, якщо на нього оформлене заовлення, та в системі для роботи адміністратора. Також заовлення зберігає дату створення, а також код та ім'я користувача, що створив заовлення.

Таблиця позицій зберігає найменшу одиницю в заовленні – певну страву. Позиція – певний набір інструкцій, що дає вичерпну характеристику персоналу кухні, що і як готувати. Основними елементами позиції є назва страви, яку потрібно приготувати, кількість цих страв, що будуть готуватися за однаковим алгоритмом (якщо вказати при оформленні позиції кількість

більше одної та зазначити особливість приготування, то буде приготована саме ця кількість страв із вказаними особливостями. Детальніше в розділі 3), комірка, що зберігає інформацію про те, чи ця страва буде вживатися в закладі, чи її потрібно упакувати для вносу (При виставленні галочки в полі «на винос», потрібно додати контейнер, або коробку, в яку буде перекладена страва після приготування), випадний список з одиничним вибором для вказання затримки при приготуванні (якщо в систему вносити одразу всі замовлені страви, то десерт бажано вказати з затримкою в 15-20 хвилин) та полем з остаточною ціною позиції. Після суми всіх полів з ціною, повна ціна замовлення змінюється в таблиці замовлень. Після оформлення замовлення дані в системі може змінити лише керівний персонал.

2.1.3 Сервер

Основним завданням серверу веб-застосунку є отримання, обробка та надсилання даних на сторінки, що вже взаємодіють з користувачем. Всі сторінки зберігаються у відповідній папці `templates` та завантажуються за допомогою методу `POST`, що дозволяє запобігти випадковому переходу по посиланню. Для запобігання накопиченню функцій, які повинна зробити сторінка при виконанні переходу та обробці внесених даних, було вирішено розробити проміжні сторінки, переходи на які користувач відслідкувати не може, а самі сторінки не відображаються в користувача взагалі, виконуючи більшу частину роботи по обробці та збереженню даних в базі. Кожен перехід між сторінками супроводжується аналізом проміжних даних, що повинні зберігатися та оновлюватися для коректної роботи клієнт-серверної архітектури та диференціації користувачів. На рисунку 2.8 зображена спрощена схема переходів між сторінками(деякі з них ще на стадії розробки) та загальний принцип роботи клієнт-серверної архітектури веб-застосунку.

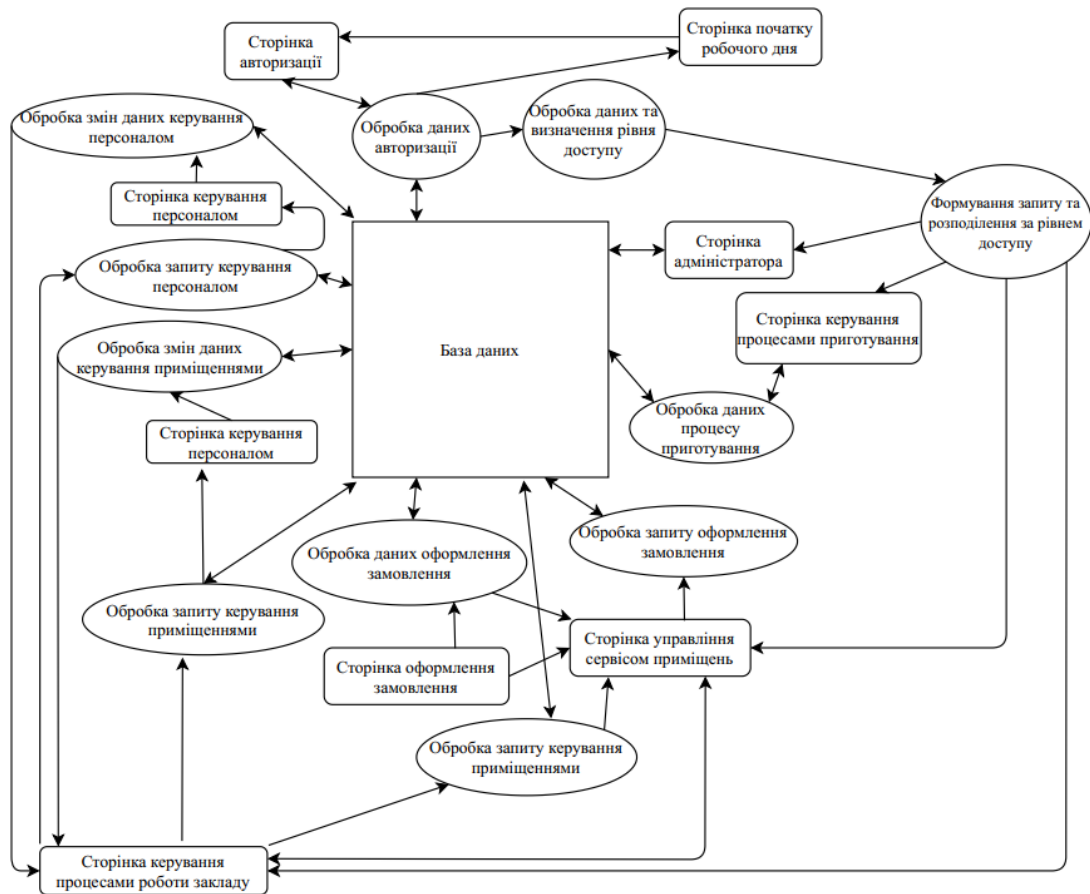


Рисунок 2.8 Спрощена схема роботи клієнт-серверної архітектури веб-сервісу

Для повного опису результатів аналізу вихідних даних важливим буде детально розібрати дані, що отримують та передають сторінки веб-застосунку. Далі буде наданий опис даних сторінок, що вже є повноцінною частиною програми.

2.1.3.1 Сторінка авторизації

Сторінка авторизації не потребує ніяких початкових даних та завантажується одразу при переході за посиланням на локальний сервер. На цій сторінці користувач вводить свій особистий пароль від системи. Далі цей пароль передається на сервер, де зашифровується та порівнюється з тими, що є у базі. Якщо паролі співпадають, з бази даних завантажується інформація про користувача та надсилається на наступну сторінку, якою може бути сторінка початку робочого дня, або прихована сторінка-аналізатор, що, згідно з інформацією по рівень доступу користувача, виконує переадресацію на відповідну сторінку.

2.1.3.2 Сторінка адміністратора

Сторінка адміністратора стає доступною лише користувачу з особливим рівнем доступу, що дозволяє йому працювати майже напяму з даними та покращувати роботу веб-застосунку шляхом його адаптації до умов роботи закладу громадського харчування. Сторінка адміністратора потребує дані користувача, а саме його персональний номер, рівень доступу та ім'я. Як таких, вихідних даних сторінка не має, адже являє собою панель адміністрації Django, що надає можливість працювати з даними, але все ж не дає адміністратору повний контроль над ними. Фактично, вихідними даними сторінки адміністратора є дані про зміни в алгоритмі роботи веб-застосунку та записи про усунення помилок під час його роботи.

2.1.3.3 Сторінка управління сервісом приміщень

Сторінка управління сервісом приміщень – основна сторінка для роботи персоналу, основним завданням якого є сервіс. Дана сторінка потребує рівень доступу не нижчий за обслуговуючий персонал, ім'я та унікальний номер користувача. Окрім цього необхідними для запуску сторінки є дані про приміщення та столи, які в них знаходяться. Крім цього потрібні дані щодо відкритих замовлень та позицій, які пов'язані з цими замовленнями. Після роботи з інтерфейсом, вихідними даними, які сторінка надсилає на сервер є обраний стіл, унікальний номер користувача, що відкрив замовлення, його рівень доступу, який все ще повинен бути не нижчим за попередній, та його ім'я.

2.1.3.4 Сторінка оформлення замовлення

Сторінка оформлення замовлення потребує рівень доступу не нижче обслуговуючого персоналу та вже стандартні ім'я та унікальний номер користувача. Крім цього, необхідними даними є вся інформація про страви та інгредієнти, номер приміщення та столу та вся інформація про попередньо зроблене замовлення, якщо воно вже було відкритим. Після роботи користувача з інтерфейсом, сторінка оформлення замовлення групує всі

додані страви в позиції та надсилає на сервер разом з ім'ям та унікальним номером користувача для ведення аудиту. Ці дані заносяться в базу, або змінюються, якщо замовлення було закритим, та вже був сплачений чек.

2.1.3.5 Сторінка керування приміщеннями

Сторінка керування приміщеннями потребує рівень доступу не нижче директора закладу, адже за допомогою цієї сторінки вносяться суттєві зміни в роботу веб-застосунку. Також необхідними є дані про користувача та інформація про усі приміщення, що є в закладі, та інформація про столи, які пов'язані з певними залами. Після роботи з інтерфейсом, дані про зміни, що були внесені в роботу приміщень надсилаються на сервер. Старі дані, які були замінені, знищуються, а нові дані записуються на вже звільнене місце. Важливо зазначити, що система не дасть змогу зберегти зміни, якщо заклад ще активно працює та в наявності є хоча б одне замовлення, що ще не закрили.

2.2 Висновки

В цьому розділі було проаналізовано розроблену стратегію створення веб-застосунку для покращення роботи закладу громадського харчування шляхом створення локального веб-сервера з клієнт-серверною архітектурою. Для докладнішого аналізу було спроектовано та створено схеми, що візуалізують складні процеси, що відбуваються при роботі веб-застосунку, розписано основні елементи архітектури та етапи створення та тестування цих елементів в середовищі, наближеному до робочого. Після аналізу зауважень та помилок, що виникли під час тестування, веб-застосунок було дороблено. Аналіз показав, що трьома найважливішими етапами створення локального веб-сервера є розробка інтерфейсу та дизайну, створення бази даних та налагодження зв'язків між таблицями та створення серверу, який виконує функцію посередника між клієнтом та базою даних, що дозволяє йому обробляти вхідну та вихідну інформацію, створювати запити до бази даних та групувати дані для надсилання відповідні сторінці клієнта.

РОЗДІЛ 3. АНАЛІЗ ФУНКЦІОНАЛУ ВЕБ-ЗАСТОСУНКУ ТА ІНСТРУКЦІЯ КОРИСТУВАЧА

В цьому розділі буде проаналізовано функціонал першої робочої версії веб-застосунку підтримки діяльності роботи закладу громадського харчування. Разом з цим буде надана інструкція користувача, адже необхідно надати повну інструкцію по користуванню веб-застосунком для надання змоги самостійного навчання стажерів роботі з системою.

3.1 Основний функціонал веб-застосунку

На даній стадії веб-застосунок знаходиться в активній стадії розробки, але існує робоча версія програмного продукту з реалізованим базовим функціоналом. Даний веб-застосунок являє собою локальний веб-сервер з клієнт-серверною архітектурою для покращення роботи закладу громадського харчування малого бізнесу, бюджет якого не розрахований на велику кількість витрат на підтримку програмного та технічного забезпечення. Користувачами даної системи є працівники закладу. Це означає, що більшу кількість часу більша кількість користувачів не матиме змогу активно взаємодіяти з системою за допомогою мишки а клавіатури, тому функціонал та дизайн розраховані на комунікацію з користувачем за допомогою екранів, що чутливі до дотику.

Реалізовані функції веб-застосунку:

- Контроль за вчасним прибуттям на робоче місце
- Формування та редагування замовлення
- Менеджмент робочих приміщень
- Контроль індивідуальної роботи робітника
- Аудит роботи системи та вдосконалення роботи системи шляхом внесення індивідуальних правок.

Далі потрібно детально розібрати функціонал кожної сторінки та тим самим створити докладу інструкцію користувачу.

3.1.1 Авторизація

Найперше вікно, з яким стикається кожен користувач – вікно авторизації. За допомогою нього персонал розпочинає робочий день, оформлює нові замовлення на себе та завершує робочий день. Основним елементом інтерфейсу вікна авторизації є поле для відображення пароля(рис. 3.1).



Рисунок 3.1 Поле для відображення введеного пароля

За допомогою клавіш, що знаходяться на екрані, користувач вводить свій пароль, що був наданий йому адміністрацією після внесення в базу персоналу. Важливо зазначити, що вся взаємодія з системою на етапі авторизації відбувається без клавіатури та мишки. Якщо користувач припуститься помилки, то він завжди може скористатися кнопкою анулювання пароля, якщо він був введений невірно, або кнопкою видалення останнього символу, якщо лише останній введений символ невірний (рис. 3.2).

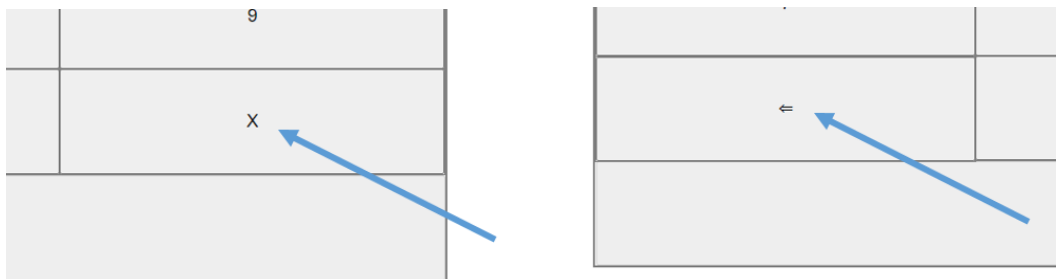


Рисунок 3.2 Кнопка анулювання поля пароля(зліва) та кнопка видалення останнього символу (справа)

Після введення персонального пароля користувачу потрібно натиснути на кнопку входження в систему(рис. 3.3). Кнопка авторизації в системі не буде

активною доти, доки не буде введена мінімальна кількість символів, що відповідає довжині пароля користувачів з найнижчим рівнем доступу.

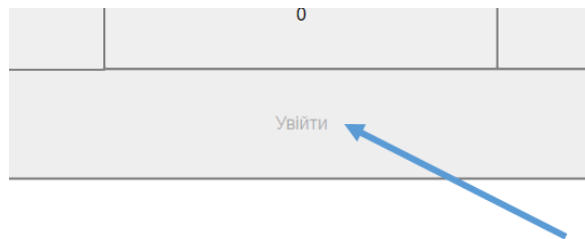


Рисунок 3.3 Кнопка підтвердження введеного пароля

При проходженні процесу авторизації користувач має три варіанти розвитку подій. Якщо введений пароль невірний, то знов відкриється сторінка авторизації. При успішному процес авторизації, якщо користувач ще не виходив на зміну, з'явиться нове меню, що сповістить про початок робочого дня(рис.3.4).

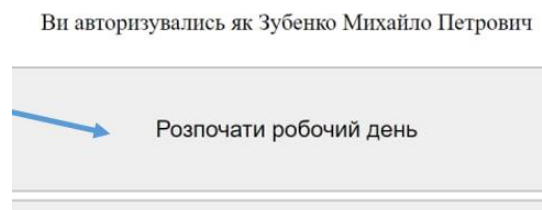


Рисунок 3.4 Кнопка для першої авторизації за зміну

Якщо користувач авторизувався не вперш за робочий день, то система його одразу направить на сторінку з відповідним рівнем доступу.

3.1.2 Сервіс столів

Сторінка сервісу столів є основною та єдиною доступною сторінкою для обслуговуючого персоналу. Ця сторінка складається з нижнього меню та таблиці розташування столів в залі. На рисунку 3.5 можна побачити, як виглядають кнопки, що відповідають за столи. Якщо в системі не існує відкритих замовлень на стіл, то вся інформація, що буде відображена – номер столу. Якщо на стіл відкрите замовлення, то на ньому окрім номера столу буде вказане ім'я користувача, який створив замовлення, та унікальний номер замовлення.



Рисунок 3.5 Кнопка столу без замовлення(зліва) та з відкритим замовленням(справа)

При натисканні на відповідну кнопку, буде відкрита сторінка оформлення замовлення. Якщо звернути увагу на верхню частину нижнього меню, то там є дві основні кнопки – завершення робочого дня та блокування(рис. 3.6). При натисканні на кнопку блокування, система перейде до сторінки авторизації, але користувач все ще буде помічений в системі як той, що знаходиться на зміні. Якщо натиснути на кнопку завершення робочого дня, система також перейде до сторінки авторизації, але змінить статус присутності користувача на зміні.

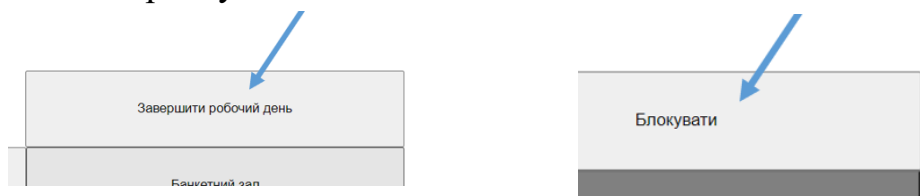


Рисунок 3.6 Кнопки блокування системи з завершенням робочого дня та без

Це зроблено для того, щоб декілька користувачів могли користуватися однією машиною, створюючи замовлення на своє ім'я, що допомагає керівному персоналу слідкувати за індивідуальною роботою кожного працівника.

Нижня частина нижньої панелі відображає набір кнопок, що дозволяють перемикатися між приміщеннями(рис.3.8). За допомогою цього можна відслідковувати навантаження на кожне приміщення окремо, та, в разі чого, направити туди людину з іншого приміщення



Рисунок 3.7 Панель приміщень

При натисканні на кнопку з назвою приміщення, вона стане сірою, а таблиця розташування столів в приміщенні відобразить розташування столів та навантаження на обране приміщення.

3.1.3 Робота з замовленням

Оформлювати замовлення та корегувати його можуть лише обслуговуючий та управлінський персонал. Сторінка роботи з замовленням складається з лівої та правої частини. Ліва частина є місцем відображення асортименту для додавання його до замовлення. У верхній частині цього меню знаходиться набір кнопок для вибору страв, напоїв та деяких інших речей, які не є стравами. (рис. 3.8)

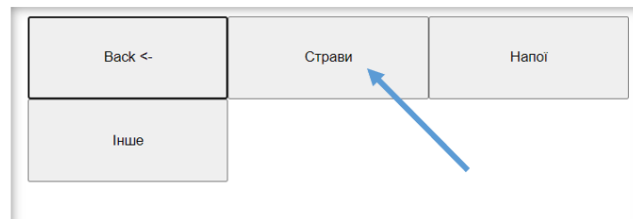


Рисунок 3.8 основні види товару

При натисканні на одну з трьох кнопок відкриється відповідне меню з категорій. Наприклад, при натисканні на страви, з'явиться можливість обрати потрібну категорію страв. Для прикладу можна обрати піцу (рис. 3.9).

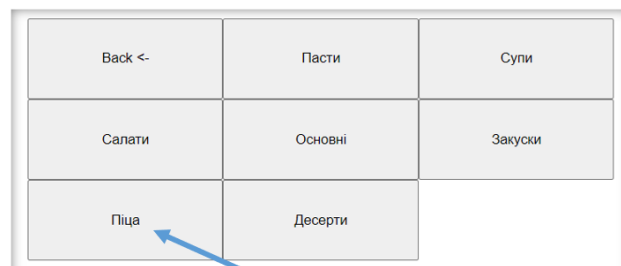


Рисунок 3.9 Категорії страв

Після натискання на категорію, в нижній частині лівого меню з'явиться набір кнопок, що дозволяють користувачу додати саме цю страву до замовлення. Для прикладу можна обрати піцу неаполітанську(рис.3.10).

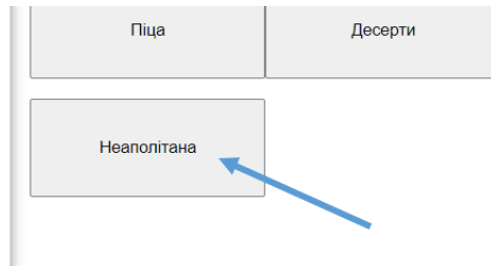


Рисунок 3.10 Страви певної категорії

При помилковому переході до категорій, або при потребі додати до замовлення страв з різних категорій, потрібно скористатися найпершою кнопкою у верхньому лівому меню(рис. 3.11). При її натисканні, вся ліва панель перетвориться на початкову.

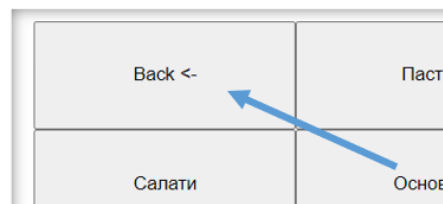


Рисунок 3.11 Кнопка відкриття таблиці видів товару

Права панель являє собою відображення даних замовлення для користувача. Вона складається з чотирьох частин. Перша частина – дані про стіл та замовлення(рис. 3.12). Якщо замовлення було відкрито щойно, то воно автоматично заповниться даними. Якщо було обрано стіл з уже відкритим замовленням, то будуть відображені дані, що були створені при відкритті замовлення.

Офіціант: Зубенко Михайло Петрович	Стіл №: 3	Гостей: 2
Замовлення № 2021052019085013	Зал: Літня тераса	Позицій: 4

Рисунок 3.12 Панель даних столу та замовлення

Наступною частиною лівої панелі є список позицій, що вже були додані та збережені в базі замовлень(рис. 3.13). Вони відображаються так само, як і нові, але повністю неактивні. У майбутньому буде додана можливість для

керуючого персоналу змінювати ці позиції. При додаванні нових позиції шляхом вибору категорії та страви в лівій панелі та збереженні їх у базі, ці позиції також стануть неактивними.

Назва	Кількість	На внос	Копи готувати	Ціна
Стейк з телятини	2	<input checked="" type="checkbox"/>	Не чекати	520
Ассорті з сирів	1	<input type="checkbox"/>	Чекати 5 хвилин	120
Салат цезар	3	<input type="checkbox"/>	Чекати 20 хвилин	360

Рисунок 3.13 Позиції, що вже були збережені в базі даних

Недавно для прикладу було обрано страву «піца неаполітана». Після цього вибору в третій частині правої панелі з'являється нова позиція з полями та списками, які можна редагувати(рис. 3.14).



Рисунок 3.14 Поле створення нової позиції

Для редагування створеної позиції використовуються деякі інструменти. Спочатку, зліва показана назва страви. Потім меню регулювання кількості. Кнопками «+» та «-» потрібно вказати кількість страв, які потрібно готувати із заданими особливостями. Якщо гість замовив дві однакові страви, та лише одну з них потрібно загорнути з собою, то потрібно створити дві окремі позиції. Наступний інструмент – встановлення прапорця. Він встановлюється тоді. Коли всі страви з даної позиції потрібно загорнути з собою. Після цього потрібно додати до замовлення спеціальний посуд, або коробку з розділу «Інше». Наступним інструментом є випадний список з вибором часу очікування. Це потрібно для того, щоб занести до системи одразу всі позиції, але не подавати їх одночасно. Наприклад, якщо гість замовив суп та десерт, то для десерту потрібно виставити затримку в 15-20 хвилин. Передостаннім елементом позиції є поле, в якому відображається ціна за позицію. Останнім інструментом є кнопка що видаляє нову позицію зі списку.

Останньою секцією правої панелі є кнопки керування замовленням(рис. 3.15).

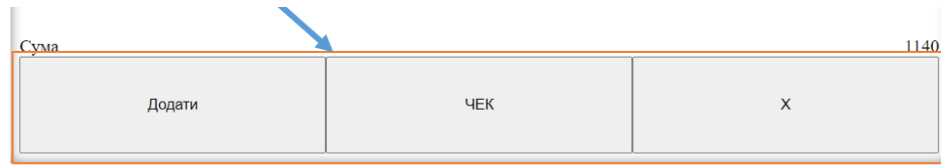


Рисунок 3.15 Панель кнопок управління позиціями замовлення

Кнопка «Додати» надсилає дані на обробку та не закриває замовлення. Його можна буде потім відкрити та продовжити додавання страв. Вже внесені та збережені страви будуть відображені у другій секції та не будуть активні. Кнопка «ЧЕК» так само як і кнопка «Додати» надсилає дані на обробку, та закриває замовлення. Її потрібно натискати тоді, коли гість попросив роздрукувати чек. Остання кнопка не зберігає зміни та повертає користувача на попередню сторінку.

3.1.4 Офіс управління

Сторінка офісу управління доступна лише для персоналу з рівнем доступу не нижче менеджера. На даний момент доступні лише три можливості управління процесами роботи закладу громадського харчування. Для переходу на сторінку певного елементу управління потрібно натиснути на потрібну кнопку(рис. 3.16).

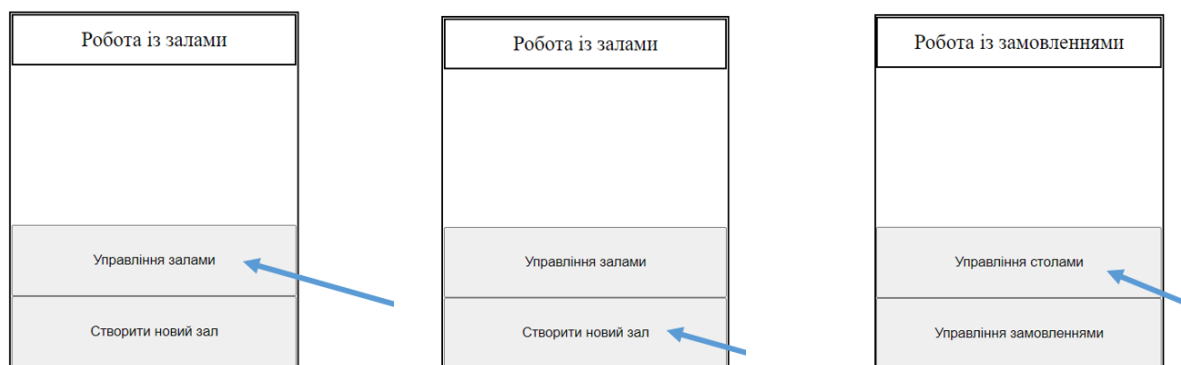


Рисунок 3.16 Кнопки управління роботи закладом

При натисканні на кнопку «Управління залами», користувач перейде до сторінки сервісу столів як керівник. В майбутньому це дасть керуючому персоналу більше інструментів роботи з замовленням. При натисканні на

кнопку управління столами, користувач буде перенаправлений на сторінку керування приміщеннями. При натисканні на кнопку створення нового залу, з'явиться нове меню(рис. 3.17).

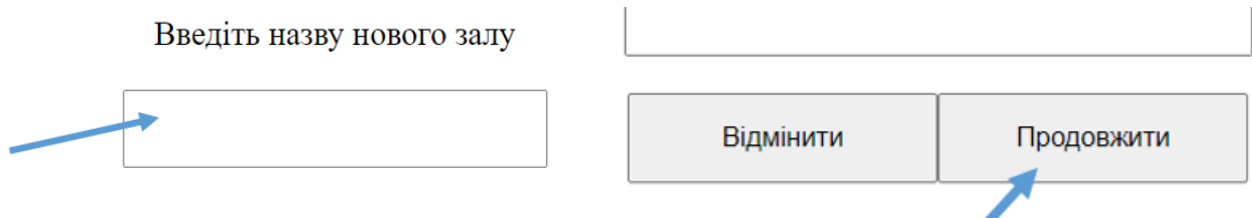


Рисунок 3.17 Створення нового залу

Меню створення нового залу – меню, яке доступне лише при використанні головного комп'ютера закладу. Для створення нового залу потрібно ввести назву нового залу та натиснути кнопку «Продовжити»

3.1.5 Менеджмент приміщень

Сторінка менеджменту приміщень доступна лише для персоналу з рівнем доступу не нижче менеджера. Сторінка розділена на дві частини – меню інструментів для роботи та таблиця редагування положення столів.

Меню інструментів складається з інструментів вибору приміщення, додавання нового столу, видалення столу, збереження даних та видалення всіх внесених змін, повертаючись до останнього збереженого варіанту(рис. 3.18).

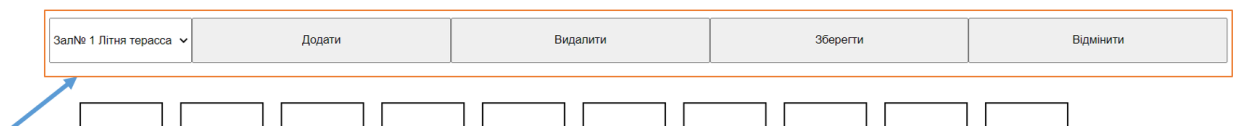


Рисунок 3.18 Панель інструментів управління приміщеннями

При натисканні на кнопку «Додати», всі можливі комірки для розміщення в них столу, підсвічуються жовтим кольором(рис. 3.19). При натисканні на підсвічену жовтим комірку, в ній створюється новий стіл з номером 1. Важливо зазначити, що зміни неможливо зберегти, якщо існує два однакових номери столів.

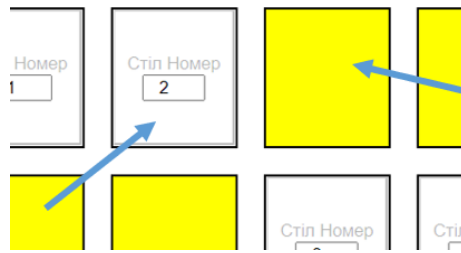


Рисунок 3.19 Комірки при виборі інструменту додавання

При натисканні на кнопку видалення, всі столи, які можливо видалити, будуть підсвічені жовтим. Після натискання на підсвічений стіл, комірка звільняється(рис. 3.20).

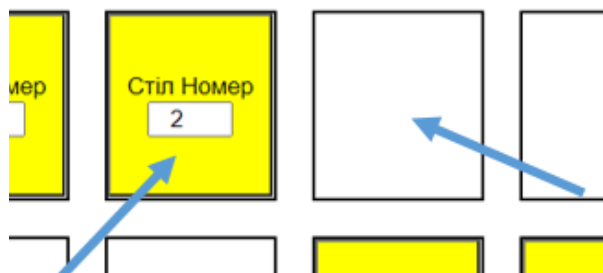


Рисунок 3.20 Комірки при виборі інструменту видалення

Важливо зазначити, що зміни в розташуванні столів в залах та зміні номерів столів неможливо зберегти, якщо є хоча б одне відкрите замовлення, або заклад громадського харчування відкритий.

3.2 Висновок

В цьому розділі було проаналізовано основний функціонал останньої на даний момент версії веб-сервісу підтримки роботи закладу громадського харчування та надано інструкцію для користувача. Були проаналізовані сторінка авторизації з усіма можливими варіантами результату роботи користувача з функціоналом сторінки, сторінка сервісу приміщень, та функціонал кнопок меню, що допомагає керуючому персоналу відслідковувати індивідуальну роботу кожного працівника, сторінка роботи з замовленням, на якій присутня можливість додавання нових позицій до замовлення та їх редагування, можливість закрити замовлення та роздрукувати чек, сторінка управління процесами роботи закладу з усім робочим функціоналом, та сторінка створення та редагування приміщень за

допомогою створення та видалення в приміщенні солів та надання їм унікальних номерів в межах одного приміщення.

ВИСНОВКИ

У даній дипломній роботі було досліджено потоки інформації серед персоналу в закладі громадського харчування та відтворено результати дослідження за допомогою схеми. Було проаналізовано основні технології для розробки локального веб-сервісу на основі клієнт-серверної архітектури. Згідно з результатами аналізу було обрано мову гіпертекстової розмітки HTML, каскадну таблицю стилів CSS, мову програмування локального веб-сервісу на стороні клієнта JavaScript, мову програмування скриптів Python, її фреймворк для створення сервера на основі базової архітектури Django та систему управління базами даних MongoDB. Також було надано результати аналізу вибору інструментів для розробки локального веб-сервісу.

Крім того, було досліджено та проаналізовано структуру приміщень у закладах громадського харчування, принципи їхньої організації роботи, градація рівня відповідальності персоналу, диференціація обов'язків персоналу різних приміщень, необхідні комунікації між персоналом різних приміщень задля ефективної роботи закладу громадського харчування та вимоги від майбутніх працівників певної посади. Результати аналізу було сформульовано та надано в першому розділі звіту.

Було описано процес розробки веб-застосунку для підтримки роботи закладу громадського харчування. Процес складається з розробки глобальної стратегії створення веб-застосунку, що включає в себе: розробку і тестування інтерфейсу та дизайну, розробки архітектури бази даних та формування зв'язків між таблицями, за допомогою вивчення принципів роботи Django, аналіз вхідних та вихідних даних зі сторінок веб-застосунку для організації неперервного зв'язку з базою даних, розробку додаткових сторінок для зменшення навантаження на сервер та тестування веб-застосунку в умовах, наближених до реальних.

Після розробки локального веб-застосунку було проаналізовано результати тестування та внесено поправки в роботу програми. Далі було надано результати аналізу функціональних можливостей веб-застосунку та

розроблено детальну інструкцію користувача на основі поточної версії програмного продукту, у яких було надано інформацію для самостійної підготовки кожного користувача до використання системи.

СПИСОК ЛІТЕРАТУРИ

1. Архіпов В., Русавська В. Організація обслуговування в закладах ресторанного господарства навч. посіб. Київ : Вид-во «Центр учбової літератури» 2009 108-117 с.
2. Бердичевский В. Х., Карсекин В. І. Технологическое проектирование предприятий общественного питания Москва: Вид-во «Высшая школа» ,1988.
3. Вільям С. Вінсент REST APIs with Django: Build powerful web APIs with Python and Django Оpubліковано самостійно 2018.
4. Дакетт Д. HTML и CSS Разработка и создание веб-сайтов: Москва: Вид-во «Эксмо» 2013.
5. Документація Django URL: <https://docs.djangoproject.com/en/3.2/> (дата звернення 27.03.2021).
6. Документація Python 3.9.5 URL: <https://docs.python.org/3/> (дата звернення 02.03.2021).
7. Кучер Л. С., Шкуратова Л. М. Организация обслуживания общественного питания: Підручник. — Москва: Вид-во «Деловая литература» 2002 – 544 с.
8. Мануал MongoDB 4.4 URL: <https://docs.mongodb.com/manual/> (дата звернення 18.04.2021).
9. Нікуленова Т.Т., Лавриненко Ю. І., Ястина Г.М. Проектирование предприятий общественного питания: Москва: Вид-во «Колос» 2000 215 с.
10. Нечаюк Л.І., Телеш Н.О. Готельно-ресторанний бізнес: менеджмент. Київ: навч. посіб. 3-е вид. Центр Навчальної Літератури 2019 59-63 с.
11. Пятницька Н. О. Організація обслуговування у закладах ресторанного господарства: навч. посіб. 2-е вид., доп. Київ: Вид-во «Центр учбової літератури» 2011 29-83 с.
12. Рейтц К., Шлюссер Т. Автостопом по Python Санкт-Петербург: Вид-во «Питер» 2018.

13. Усіна А. І., Давидова О. Ю., Сегеда І. В., Кононенко Т. П. Організація послуг харчування навч. посіб. Харків: ХНУМГ 2014 32-44 с.
14. Хавербеке М. Выразительный JavaScript: 3-е вид. Москва: Вид-во «Питер Пресс» 2019.