

(повна назва факультету)

Комп'ютерних наук

(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: **Оптимізація процесів управління проєктами у сфері  
обслуговування на прикладі закладу харчування**

Виконав(ла): студент(ка) 6 курсу, групи СНМ-61  
спеціальності 122 – комп'ютерні науки

(шифр і назва спеціальності)

	_____	<b>Войтович А. І.</b>
	(підпис)	(прізвище та ініціали)
Керівник	_____	<b>Марценко С. В.</b>
	(підпис)	(прізвище та ініціали)
Нормоконтроль	_____	<b>Мацюк О. В.</b>
	(підпис)	(прізвище та ініціали)
Завідувач кафедри	_____	<b>Боднарчук І. О.</b>
	(підпис)	(прізвище та ініціали)
Рецензент	_____	<b>Томашевський Б. П.</b>
	(підпис)	(прізвище та ініціали)

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)  
Кафедра Комп'ютерних наук  
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

« \_\_\_\_\_ » \_\_\_\_\_  
(підпис) Боднарчук І. О.  
(прізвище та ініціали)  
20\_\_р.

**З А В Д А Н Н Я**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня магістр  
(назва освітнього ступеня)  
за спеціальністю 122- Комп'ютерні науки  
(шифр і назва спеціальності)  
студенту Войтович Анастасії Ігорівні  
(прізвище, ім'я, по батькові)

1. Тема роботи Оптимізація процесів управління проєктами в сфері обслуговування на прикладі закладу харчування

Керівник роботи Марценко Сергій Володимирович к.т.н доцент кафедри комп'ютерних наук  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « \_\_\_\_\_ » \_\_\_\_\_ 2021 року № \_\_\_\_\_

2. Термін подання студентом завершеної роботи \_\_\_\_\_

3. Вихідні дані до роботи \_\_\_\_\_

4. Зміст роботи (перелік питань, які потрібно розробити)

1. Методи управління проєктами для сфери обслуговування. 2. Вибір методу управління за допомогою MAI. 3. Розробка системи для оптимізації роботи кав'ярні. 4. Охорона праці та безпека в надзвичайних ситуаціях.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

Оптимізація процесів управління проєктами в сфері обслуговування на прикладі закладу харчування. Управління проєктами в закладах харчування. Огляд методів управління. Вибір методу управління проєктами за допомогою MAI. Результати обчислень.

**6. Консультанти розділів роботи**

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Дмитроца Л. П. доцент		
Безпека в надзвичайних ситуаціях	Клепчик В. М. ст. викладач		

7. Дата видачі завдання 27 вересня 2021р**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Ознайомлення із завданням до кваліфікаційної роботи		Виконано
2	Підбір наукових джерел по темі		Виконано
3	Обґрунтування актуальності дослідження		Виконано
4	Аналіз предмету дослідження та предметної області		Виконано
5	Оформлення розділу «Методи управління проектами для сфери обслуговування.»		Виконано
6	Оформлення розділу «Вибір методу управління за допомогою МАІ»		Виконано
7	Оформлення розділу «Розробка системи для оптимізації роботи кав'ярні»		Виконано
8	Виконання завдання до підрозділу «Охорона праці»		Виконано
9	Виконання завдання до підрозділу «Безпека в надзвичайних ситуаціях»		Виконано
10	Оформлення кваліфікаційної роботи		Виконано
11	Проходження нормоконтролю		Виконано
12	Проходження перевірки на плагіат		Виконано
13	Попередній захист кваліфікаційної роботи		Виконано
14	Захист кваліфікаційної роботи		

Студент

\_\_\_\_\_  
(підпис)Войтович А. І.  
(прізвище та ініціали)

Керівник роботи

\_\_\_\_\_  
(підпис)Марценко С. В.  
(прізвище та ініціали)

## АНОТАЦІЯ

"Оптимізація процесів управління проектами в сфері обслуговування на прикладі закладу харчування" // Войтович Анастасія Ігорівна // Тернопільський національний технічний університет ім. І. Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНм-61 // Тернопіль, 2021 // с. – 73, рис. – 26, табл. – 11, джерел – 60.

Ключові слова: ОПТИМІЗАЦІЯ, SCRUM, KANBAN, SIX SIGMA, КЛАСИЧНИЙ ПРОЕКТНИЙ МЕНЕДЖМЕНТ, МЕТОД АНАЛІЗУ ІЄРАРХІЙ.

У магістерській роботі методологія вдосконалення управління проектами в сфері обслуговування. Огляд застосування відомих методів управління таких як, класичний проектний менеджмент, scrum, kanban, six sigma. Вибір кращого методу за допомогою методу аналізу ієрархій. Ця робота демонструє як за допомогою МАІ вибрати найкращий метод відповідно до ваших потреб. Також робота демонструє застосування класичного проектного менеджменту для оптимізації роботи закладу харчування.

## ANNOTATION

"Optimization of project management processes in the field of service on the example of a catering establishment" // Voitovych Anastasia Igorivna // Ternopil National Technical University. I. Pulyuya, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, CHM-61 Group // Ternopil, 2021 // p. - 73, Fig. – 26, table. -11, sources -60.

Keywords: OPTIMIZATION, SCRUM, KANBAN, SIX SIGMA, CLASSICAL PROJECT MANAGEMENT, METHOD OF ANALYSIS OF HIERARCHIES.

In the master's thesis the methodology of improving project management in the field of service. Overview of the application of known management methods such as, classical project management, scrum, kanban, six sigma. Choosing the best method using the method of hierarchy analysis. This work demonstrates how to use MAI to choose the best method according to your needs. The work also demonstrates the use of classical project management to optimize the work of the catering establishment.

## ЗМІСТ

	ВСТУП .....	7
1	МЕТОДИ УПРАВЛІННЯ ПРОЄКТАМИ ДЛЯ СФЕРИ ОБСЛУГОВУВАННЯ	9
	1.1 Опис предметної області .....	9
	1.1.1 Роль процесів управління для реалізації проєктів.....	9
	1.1.2 Елементи процесу управління проєктом для закладів харчування .....	13
	1.2 Опис методів управління проєктами .....	15
	1.2.1 Класичний проєктний менеджмент .....	15
	1.2.2 Scrum .....	19
	1.2.3 Kanban .....	21
	1.2.4 Six Sigma .....	23
2	ВИБІР МЕТОДУ УПРАВЛІННЯ ЗА ДОПОМОГОЮ МАІ	26
	2.1 Огляд методу аналізу ієрархій.....	26
	2.1.1 Відбір проєкту за стратегічними цілями компанії .....	27
	2.1.2 Процес аналітичної ієрархії.....	30
	2.1.3 Використання МАІ для оцінки та відбору проєкту.....	31
	2.1.4 Приклад використання МАІ для вибору кращого проєкту .....	35
	2.1.5 Програмне забезпечення MS EXCEL для процесу МАІ.....	40
	2.1.6 Моделювання критеріїв.....	41
	2.2 Вибір кращого методу управління використовуючи МАІ .....	45
3	РОЗРОБКА СИСТЕМИ ДЛЯ ОПТИМІЗАЦІЇ РОБОТИ КАВ'ЯРНІ .....	50
4	Охорона праці та безпека в надзвичайних ситуаціях .....	58

4.1 Організація охорони праці в закладах громадського харчування .....	58
4.2 Шкідливий вплив іонізуючого випромінювання .....	61
ВИСНОВКИ.....	67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	68
ДОДАТКИ	

## ВСТУП

**Актуальність теми.** Заклади харчування являються не від'ємною частиною сучасного суспільства. Але мало хто задумується про сферу хореки, як про цілісну структуру. А власне структуру, в якій всі деталі є важливими, потребують детальної уваги та мають взаємозв'язок. Тобто приготування страв, управління продажами, культура обслуговування, рекламна та цінова політика є єдиною складовою. Успішний заклад харчування можна впізнати за декількома ознаками:

- основні бізнес-процеси підприємства є формалізованими та налагодженими;
- управління підприємством є активним, постійно націленим на інновації, ґрунтується на вибраній стратегії;
- управління закладом є автоматизованим;
- кваліфікований та клієнто-орієнтований персонал;
- наявна стратегія підприємства, операційне та фінансове планування відбувається за стратегічним планом.

Функціонування закладів харчування представляють набором бізнес-процесів різних рівнів. Проте наразі, залишається відкритим питання управління бізнес-процесами, як проєктами в ресторанному бізнесі.

У зв'язку з важливістю та актуальністю теми було зроблено безліч досліджень та публікацій.

**Мета роботи.** Метою роботи є проаналізувати актуальність процесів управління проєктами в сфері обслуговування, та вплив управління на розвиток закладів обслуговування, а власне закладів харчування.



Виходячи з мети дослідження, описаної вище, ми визначили наступні питання дослідження:

- Які є системи управління проєктами?
- Як обрати систему управління за допомогою MAI?
- Як адаптувати вибрану систему для управління проєктами в вибраному закладі?

**Об'єкт дослідження:** процеси управління проєктами в сфері обслуговування.

**Предмет дослідження:** методи управління проєктами.

**Методи дослідження:** Для досягнення мети кваліфікаційної роботи використовувались:

- Метод аналізу ієрархій за допомогою, якого ми вибрали метод управління проєктами. Ми розглянули чотири методи управління проєктами і на основі цих методів за допомогою MAI ми вибрали метод який найкраще підходить для управління в сфері обслуговування, а саме в закладах харчування

**Наукова новизна отриманих результатів.** Наукова новизна полягає у вирішенні задачі застосування автоматизованого методу вибору методу управління закладом харчування на основі методу аналізу ієрархій.

**Практичне значення отриманих результатів.** За допомогою отриманих результатів ми розробили веб-застосунок, за допомогою якого, відбувається оптимізація роботи закладу харчування отриманим за розрахунками методом.

**Апробація результатів та особистий внесок здобувача.** Основні положення роботи доповідались, розглядались та обговорювались на науковій конференції Тернопільського національного технічного університету. Результати кваліфікаційної роботи опубліковані у тезах студентської наукової конференції, яка проводилась у ТНТУ.

# 1 МЕТОДИ УПРАВЛІННЯ ПРОЄКТАМИ ДЛЯ СФЕРИ ОБСЛУГОВУВАННЯ

## 1.1 Опис предметної області

### 1.1.1 Роль процесів управління для реалізації проєктів

Чи могли Велику Китайську стіну, піраміди чи Стоунхендж, побудувати без управління проєктом? Можна сказати, що концепція управління проєктами існує з самого початку історії. Це дозволило керівникам планувати сміливі та масштабні проєкти та керувати фінансуванням, матеріалами та робочою силою у визначені терміни.

Наприкінці 19 століття в Сполучених Штатах масштабні урядові проєкти були поштовхом для прийняття важливих рішень, які стали основою для методології управління проєктами, наприклад, трансконтинентальної залізниці, яку почали будувати в 1860-х роках. Несподівано керівники підприємств опинилися перед складним завданням організації ручної праці тисяч робітників, а також переробки та збирання безпрецедентної кількості сировини.

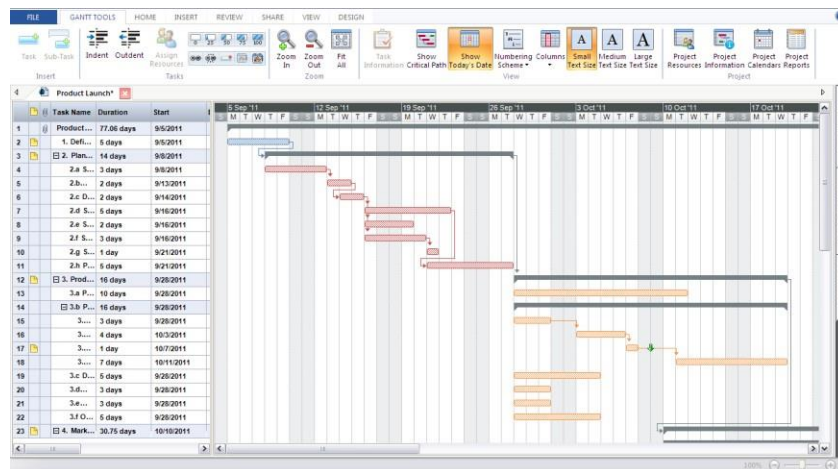


Рисунок 1.1 – Діаграма Ганта MindView

Генрі Гант, дуже детально вивчив порядок операцій у роботі і найбільш відомий розробкою діаграми Ганта в 1910-х роках. Діаграма Ганта (див. рис. 1.1) – це популярний тип стовпчастої діаграми, яка ілюструє графік проектута стала загальноприйнятою технікою представлення етапів та заходів проекту, щоб їх могла зрозуміти широка аудиторія. Незважаючи на те, що зараз є загальноприйнятою технікою створення діаграм, діаграми Ганта вважалися революційними на момент їх появи. Діаграми Ганта використовувалися у великих інфраструктурних проєктах у Сполучених Штатах, включаючи грему Гувера та систему міждержавних автомагістралей, і досі вважаються важливими інструментами в управлінні проєктами.

До середини 20-го століття проєктами керували на тимчасовій основі, використовуючи переважно діаграми Ганта та неформальні методи та інструменти. У той час був започаткований Манхеттенський проєкт, і його складність була можлива лише завдяки методам управління проєктом. Манхеттенський проєкт – кодова назва, дана зусиллям союзників розробити першу ядерну зброю під час Другої світової війни. У ньому задіяно понад 30 різних проєктних майданчиків у Сполучених Штатах і Канаді, а також тисячі персоналу зі Сполучених Штатів, Канади та Великобританії. У Манхеттенському проєкті, народженому в результаті невеликої дослідницької програми, яка розпочалася в 1939 році, в кінцевому підсумку буде зайнято 130 000 людей, вартість на загальну суму майже 2 мільярди доларів США, що призвело до створення кількох виробничих і дослідницьких майданчиків, які функціонували таємно. У 1945 році в рамках проєкту вдалося розробити та підірвати три одиниці ядерної зброї.

1950-ті роки стали початком сучасної ери управління проєктами. Розроблено дві математичні моделі планування проєкту.

Методика оцінки та огляду програми (PERT) була розроблена Буз-Алленом і Гамільтоном в рамках програми ракетних підводних човнів ВМС США Polaris. PERT – це в основному метод аналізу завдань, пов’язаних із завершенням проєкту, особливо часу, необхідного для виконання кожного завдання, залежностей між завданнями та мінімального часу, необхідного для завершення всього проєкту (рис. 1.2).

Метод критичного шляху (CPM) був розроблений у спільному підприємстві DuPont Corporation і Remington Rand Corporation для управління проєктами технічного обслуговування заводів. Критичний шлях визначає float, або гнучкість графіка, для кожного виду діяльності, обчислюючи найранішу дату початку, найранішу дату завершення, останню дату початку та останню дату завершення для кожної дії. Критичний шлях, як правило, є найдовшим повним шляхом у проєкті. Будь-яка активність з часом float, що дорівнює нулю, вважається завданням критичного шляху. CPM може допомогти вам з’ясувати, скільки часу знадобиться на завершення вашого складного проєкту, а також які дії є критичними, тобто їх потрібно виконати вчасно, інакше весь проєкт займе більше часу. Ці математичні методи швидко поширилися на багатьох приватних підприємствах.

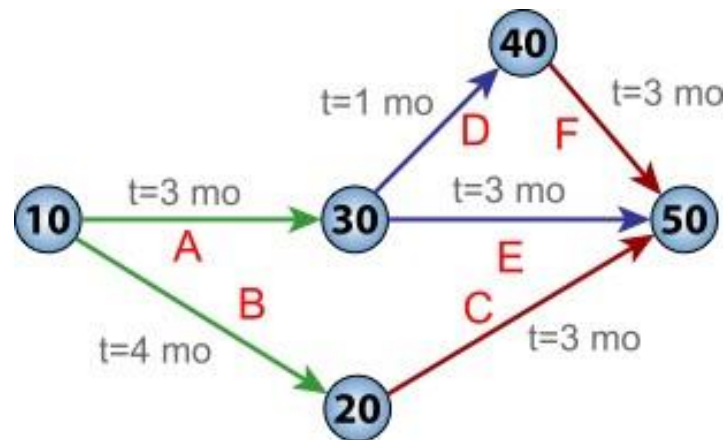


Рисунок 1.2 – PERT-діаграма

Управління проектами в його нинішньому вигляді почало приживатися кілька десятиліть тому. На початку 1960-х років промислові та ділові організації почали розуміти переваги організації роботи навколо проектів. Вони розуміли критичну потребу в спілкуванні та інтеграції роботи в різних відділах і професіях.

Навички, отримані під час вивчення менеджменту проектів, можна використовувати в більшості кар'єр, а також у вашому повсякденному житті. Сильні навички планування, хороша комунікація, здатність реалізувати проєкт для доставки продукту чи послуги, а також моніторинг ризиків та управління ресурсами забезпечать перевагу на шляху до вашого успіху. Керівників проєктів

можна побачити в багатьох галузях, включаючи сільське господарство та природні ресурси; мистецтво, ЗМІ та відпочинку; будівельні справи та будівництво; енергетика та комунальні послуги; інжиніринг і проєктування; мода

та інтер'єри; фінанси та бізнес; охорона здоров'я та обслуговування людей; гостинність, туризм та відпочинок; виробництво та розробка продукції; державні та приватні освітні послуги; державні послуги; роздрібна та оптова торгівля; транспортування; та інформаційні технології.

Власники бізнесу обов'язково повинні мати певні навички управління проектами. У будь-якому успішному бізнесі продукт або послуга, що надаються клієнту, задовольняють його потреби багатьма способами. Товар або послуга мають бажану якість, витрати відповідають очікуванням споживача, а своєчасність продукту чи послуги відповідає кінцевому терміну для покупця цього товару.

Основами управління проектами є надання продукту/послуги згідно з графіком, вартістю, обсягом та вимогами до якості. Власники бізнесу потребують навичок планування, організації та визначення обсягу, а також вміння аналізувати, спілкуватися, складати бюджет, персонал, обладнати, впроваджувати та виконувати.

Розуміння фінансів, операцій та витрат бізнесу є одними з навичок, яким навчаються та практикують керівники проєктів. Деякі підприємства можуть більше зосередитися на бухгалтерському обліку, наданні фінансових консультацій, продажах, навчанні, зв'язках з громадськістю, а також на ролі актуарія або логіста. Власники бізнесу можуть мати туристичне агентство або надавати гостинність. Власники бізнесу можуть керувати вітриною магазину або місцем на ринку свого міста.

### **1.1.2 Елементи процесу управління проєктом для закладів харчування**

Розглянемо, як відбувається управління проєктом на прикладі закладу харчування.

Менеджери ресторану відповідають за щоденну роботу ресторану, який готує та подає страви та напої клієнтам. Сильні навички планування, особливо координація роботи з різними відділами (кухня, їдальня, банкетні операції, менеджери громадського харчування, постачальники, що надають товари),

гарантують, що клієнти будуть задоволені своїм обідом. Здатність менеджерів залучати й утримувати працівників, а також контролювати ефективність і навчання працівників забезпечують якість із стримуванням витрат. Планування в багатьох аспектах, не тільки для персоналу, але й у терміни доставки продуктів харчування, має вирішальне значення для задоволення очікувань клієнтів.

Управління ризиками є важливим для забезпечення безпеки та якості харчових продуктів. Менеджери контролюють замовлення на кухні, щоб визначити, де можуть виникнути затримки, і вони співпрацюють з шеф-кухарем, щоб запобігти цим затримкам. Для того, щоб ресторан залишався відкритим, важливо дотримуватися законодавчих норм, тому керівники ресторану керують прибиранням обідніх зон та миттям столового посуду, кухонного начиння та обладнання. Вони забезпечують дотримання стандартів безпеки та законності, особливо при подачі алкоголю. Чутливість і сильні комунікативні навички необхідні, коли клієнти скаржаться або працівники відчують тиск, оскільки приходить більше клієнтів, ніж передбачалося.

Фінансові знання необхідні для надійного функціонування ресторану, особливо для відстеження спеціальних проєктів, подій та витрат на різноманітні меню. Безперебійне проведення заходів з громадського харчування може бути результатом використання планів проєкту та філософії управління проєктами. Менеджер ресторану або шеф-кухар аналізують рецепти, щоб визначити витрати на їжу, робочу силу та накладні витрати; визначає розмір порції та поживність кожної порції; і призначає ціни на різні пункти меню, щоб товари можна було замовити та отримати вчасно.

Планування є запорукою успішної реалізації. Менеджери або шеф-кухари повинні оцінювати потреби в їжі, розміщувати замовлення у дистриб'юторів і планувати доставку свіжих продуктів і запасів. Вони також планують регулярні

послуги (обслуговування обладнання, боротьба з шкідниками, вивезення відходів) та доставку, включаючи послуги прання скатертин або інтенсивне прибирання залу чи кухонного обладнання, що відбуватися коли заклад зачинений. Успішний ресторан покладається на багато чинників, на яких наголошує професія управління проєктами.

## **1.2 Опис методів управління проєктами**

В подальшому у даній роботі, буде описано розробку автоматизації роботи закладу харчування, а саме оптимізацію швидкості та якості обслуговування клієнтів, для цього нам потрібно вибрати метод управління нашим проєктом. В даному розділі буде описано 4 методи управління проєктами, для подальшого вибору одного, найбільш ефективного методу.

### **1.2.1 Класичний проєктний менеджмент**

Перший метод який ми розглянемо це класичний проєктний менеджмент.

Цей метод полягає в тому, що кожен проєкт проходить один і той же життєвий цикл, який включає п'ять етапів: ініціювання, планування, виконання, контроль і закриття.



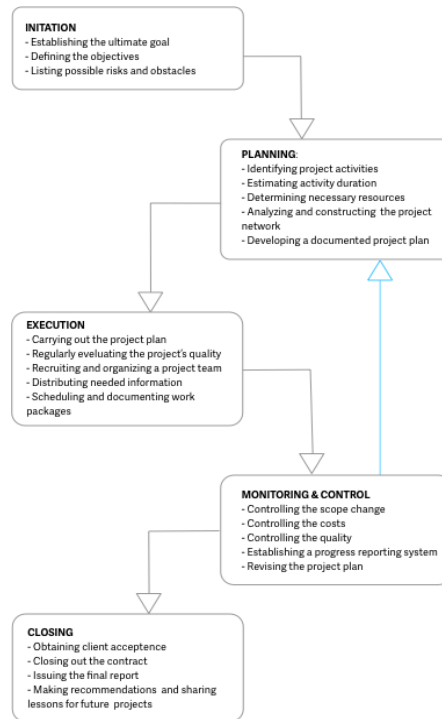


Рисунок 1.3 – зображення п'яти етапів класичного проєктного менеджменту

Для того щоб переконатись, що всі завдання виконуються в заздалегідь визначеній послідовності використовують діаграму Ганта.

Діаграма Ганта є найважливішою технікою в традиційному управлінні проєктами. Його творцем був Генрі Гант, тому його вважають батьком традиційного управління проєктами. Діаграма Ганта дає простий огляд проєкту. Це один із найкорисніших способів представлення завдань та діяльності проєкту на часовій шкалі.

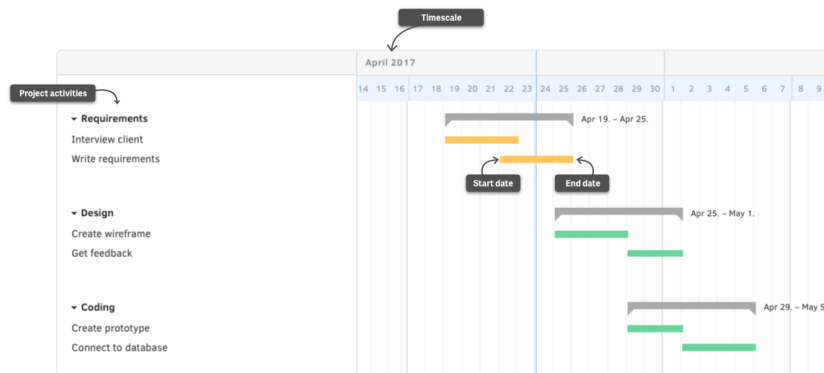


Рисунок 1.4 – Діаграма Ганта

Ліва частина діаграми Ганта показує діяльність проєкту, а верхня – часову шкалу. Кожна діяльність представлена панеллю. Позиція рядка говорить нам про початок, тривалість та кінець дії.

Діаграма Ганта в основному використовується, щоб відстежувати графіки проєктів і зробити управління проєктом менш напруженим. Діаграма Ганта допомагає нам чіткіше зрозуміти взаємозв'язок між завданнями, тримати всіх членів команди на одній сторінці та успішно завершити проєкт.

Існує багато обмежень для проєкту, і три найпоширеніших – це час, вартість та обсяг. Вони є частиною кожного проєкту і разом становлять трикутник управління проєктом.[22]

Час є безцінним ресурсом. Хоча ми можемо контролювати процеси та вносити необхідні зміни для їх покращення, ми не можемо контролювати час. Одна з найбільших проблем, з якою має зіткнутися кожен керівник проєкту, – це ефективно використовувати час, тримати проєкт за графіком і досягати бажаних цілей.

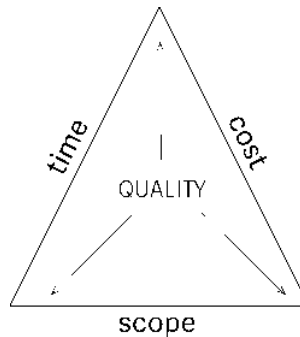


Рисунок 1.5 – Трикутник управління проектом

Як керівник проекту, ви повинні визначити бюджет на ранній стадії проекту, а потім порівняти його з цифрою, яку спочатку запропонував ваш клієнт. Якщо клієнт вирішить витратити певну суму грошей, яка не відповідає вашим вимогам, ви можете підготувати бізнес-пропозицію, яка включатиме оцінку загальної вартості проекту. Ця пропозиція допомагає клієнту базувати своє рішення на більш точних оцінках.

Всі три обмеження взаємопов'язані і сильно залежать одне від одного. Після скорочення часу, виділеного на проект, вартість збільшується. Крім того, обсяг проекту диктує темп і кількість ресурсів, необхідних для реалізації та успішного завершення проекту.

Площа всередині трикутника відображає якість і є кінцевою метою виконання кожного проекту. Хороший менеджер проекту знайде спосіб контролювати всі три обмеження трикутника і виробляти найкращу якість.

Кінцева мета менеджера проекту – задовольнити вимоги своїх клієнтів і відповідати стандартам якості. Для цього ви, як керівник проекту, повинні контролювати як якість, так і обмеження.[20]

На основі традиційного управління проектами: управління якістю

Щоб підвищити якість управління продуктом і проектом, вам слід інтегрувати такі інструменти:

- Постійне управління якістю
- Управління якістю процесу

Постійне управління якістю використовується для аналізу будь-яких прогалів, у яких можна було б виконати певні процеси та кроки для покращення якості. Ви можете здійснити будь-яку кількість покращень, а потім звірити їх із цільовими покращеннями. Він циклічний і в основному присутній в організаціях, орієнтованих на клієнтів.

Управління якістю процесу включає фактори, які мають значний вплив на успіх бізнес-процесів, і ґрунтується на припущенні, що організація задокументувала свою місію та бачення.

Використовуючи систему оцінки, усі процеси, які є критичними для успішного бізнесу, визначаються, аналізуються та пов'язані з безперервним управлінням якістю.[25] Результати аналізу допомагають менеджеру проекту внести необхідні покращення для заданих процесів, і наступним етапом є ініціювання життєвого циклу управління проектом.

### **1.2.2 Scrum**

Наступний метод це Scrum. Scrum – це методологія управління проектами, яка пропонує принципи та процеси для покращення виконання. У рамках розробки програмного забезпечення методологія Scrum є однією з найпопулярніших і простих фреймворків для застосування принципів Agile на практиці.

Метою Scrum є покращення комунікації, командної роботи та швидкості розвитку. Якщо ви чуєте, як люди говорять про такі аспекти, як спринти, сутички,

відставання та вигорання, вони, ймовірно, говорять про Scrum або щось похідне від нього.

Scrum насправді не методологія управління проектами, а основа для постійної розробки та підтримки складних продуктів. Scrum – це легкий підхід, який визначає простий набір ролей, зустрічей, які називаються Scrum церемоніями, та інструментів для ефективного, ітераційного та поступового надання цінних функціональних можливостей, які можна передавати.

По суті, Scrum має на меті надання повноважень Scrum команді, яка самостійно керує, визначає ролі та обов'язки, щоб створити здорову напругу між якнайшвидшою доставкою правильних речей у правильний спосіб.

Scrum підтримує використання невеликої багатофункціональної команди до 9 людей, які працюють над елементами в резерві (набір історій або вимог користувачів), які були визначені та визначені власником продукту.



Рисунок 1.6 – Схема процесу Scrum

Робота розділена на «спринти», цикл розробки, як правило, 2-4 тижні, протягом якого щодня відбуваються «Scrums», де команда повідомляє про прогрес і перешкоди. Наприкінці кожного спринту робота перевіряється на нараді з огляду спринту, щоб разом з Власником продукту визначити, чи відповідає вона вимогам (DoD).

Скрам фасилітує та обслуговує Скрам-майстер. Вони очолюють Scrums, спринти, демонстрації та огляди, а також «ретроспективу спринту» після кожного спринту, щоб забезпечити постійну оптимізацію та вдосконалення команди.[27]

Спочатку Scrum був розроблений для розробки програмного забезпечення. Таким чином, хоча є Agile артефакти від Scrum, хоча їх можна використовувати, Scrum не вписується в типовий більш стратегічний і креативний світ агентства. Навіть у веб-проєктах агентств фіксовані бюджети, терміни та обсяг не забезпечують гнучкості для самокерованої команди Scrum, у проєкті з визначеним початком і кінцем.

Також багато команд використовують деякі концепції Scrum – самоорганізацію, міжфункціональні команди, щоденні стендапи, демонстрації прогресу та ретроспективи та використовують їх у якомусь гібридному підході.

### **1.2.3 Kanban**

Kanban – це методологія управління проєктами, орієнтована на принципи Lean і суворий процес підвищення ефективності. Багато в чому він схожий на Scrum – це пов’язано з достроковим випуском і часто разом із командою, яка співпрацює й самостійно керує. Але порівняно зі Scrum, Kanban – це більш еволюційна зміна, м’яка посадка у світ Agile, оскільки він менш директивний.

Методологія Kanban – це легкий процес, гнучкий, не має визначених ролей і просто намагається підвищити пропускну здатність, зосередивши команду на

справді важливих речах.[25] Основними методами є візуалізація робочого процесу, обмеження незавершеної роботи, вимірювання часу виконання, чітка політика процесу та постійна оцінка можливостей покращення.

Канбан зосереджується на роботі, яка постійно випускається, швидше та з кращою якістю. Він чудово підходить для операційного середовища або середовища, де пріоритети можуть часто змінюватися. Канбан зосереджується на вимірюванні часу виконання – скільки часу потрібно після інструктажу для доставки.

За допомогою Канбан керівники проєктів зазвичай використовують наклейки на дошці Канбан (наприклад, дошці або онлайн-інструмент, як-от Trello) щоб представити робочий процес команди з такими простими категоріями, як «Завдання», «Виконання» та «Готово».



Рисунок 1.7 – Схема процесу Канбан

Це візуалізує те, що ви хочете зробити, і обмежує незавершену роботу (WIP), так що хід роботи покращується, коли ви вимірюєте й оптимізуєте середній час виконання елементів.

Це також дає команді візуальне відображення того, що буде далі, що дозволяє легко змінити пріоритети, виявити проблеми процесу та запобігти зупинці завдань. Це також допомагає їм побачити, як будь-яке нове завдання може вплинути на поточну роботу.

Kanban добре підходить для роботи, яка вимагає стабільного результату, як-от виробництво або підтримка та обслуговування.

### **1.2.4 Six Sigma**

Six Sigma – це набір інструментів і методів управління, призначених для покращення бізнесу шляхом зниження ймовірності помилок. Це підхід, орієнтований на дані, який використовує статистичну методологію для усунення дефектів.[30]

Концепція Six Sigma має просту мету – надання майже ідеальних товарів і послуг для трансформації бізнесу для оптимального задоволення клієнтів (CX).

Шість сигм базується на п'яти ключових принципах:

#### **1. Орієнтація на клієнта**

Це ґрунтується на поширеному переконанні, що «замовник – король». Основна мета – принести максимальну користь клієнту. Для цього бізнес повинен розуміти своїх клієнтів, їхні потреби та те, що стимулює продажі чи лояльність. Це вимагає встановлення стандарту якості, який визначається вимогами клієнта або ринку.

#### **2. Виміряйте потік цінності та знайдіть свою проблему**

Напишіть етапи певного процесу, щоб визначити зони відходів. Зберіть дані, щоб виявити конкретну проблемну область, яку потрібно вирішити або змінити. Мати чітко визначені цілі збору даних, включаючи визначення даних, які потрібно зібрати, причину збору даних, очікуване розуміння, забезпечення



точності вимірювань та встановлення стандартизованої системи збору даних. Перевірте, чи допомагають дані досягненню цілей, чи потрібно їх уточнювати, чи потрібно зібрати додаткову інформацію. Визначте проблему. Поставте запитання та знайдіть першопричину.

### 3. Позбавтеся від сміття

Як тільки проблема буде виявлена, внесіть зміни до процесу, щоб усунути варіації, таким чином усуваючи дефекти. Видаліть у процесі дії, які не додають цінності клієнта. Якщо потік створення цінності не розкриває, де криється проблема, використовуються інструменти, які допомагають виявити відхилення та проблемні області. Оптимізуйте функції для досягнення контролю якості та ефективності. Зрештою, видаляючи вищезгаданий мотлох, усуваються вузькі місця в процесі.

### 4. Продовжуйте крутити м'яч

Залучайте всіх зацікавлених сторін. Прийміть структурований процес, коли ваша команда вносить свій внесок і співпрацює зі своїм різноманітним досвідом для вирішення проблем.

Процеси «Шість сигм» можуть мати великий вплив на організацію, тому команда повинна володіти принципами та методологіями, які використовуються. Таким чином, для зниження ризику невдач проекту чи перепроєктування та забезпечення оптимальної роботи процесу необхідні спеціальні навчання та знання.

### 5. Забезпечте гнучку та чуйну екосистему

Суть Six Sigma – це трансформація та зміни бізнесу. Коли несправний або неефективний процес усувається, це вимагає зміни практики роботи та підходу співробітників. Надійна культура гнучкості та чуйності до змін у процедурах може забезпечити спрощену реалізацію проекту. Залучені люди та відділи

повинні мати можливість легко адаптуватися до змін, тому, щоб полегшити це, процеси мають бути розроблені для швидкого та безперервного впровадження. Зрештою, компанія, яка зосереджується на даних, періодично перевіряє підсумки та за необхідності коригує свої процеси, може отримати конкурентну перевагу.

Методологія шести сигм.

Дві основні методології Six Sigma – DMAIC і DMADV. Кожен із них має власний набір рекомендованих процедур, які слід запровадити для трансформації бізнесу.

DMAIC – це метод, керований даними, який використовується для покращення існуючих продуктів або послуг для кращого задоволення клієнтів. Це аббревіатура для п'яти етапів: D – визначення, M – вимірювання, A – аналіз, I – покращення, C – контроль. DMAIC застосовується при виготовленні продукту або наданні послуги.

DMADV є частиною процесу Design for Six Sigma (DFSS), який використовується для проєктування або перепроєктування різних процесів виробництва продукції або надання послуг. П'ять етапів DMADV: D – визначення, M – вимірювання, A – аналіз, D – проєктування, V – перевірка. DMADV використовується, коли існуючі процеси не відповідають умовам замовника, навіть після оптимізації, або коли потрібно розробити нові методи.

В наступному розділі ми будемо порівнювати дані методології. І також виберемо найкращу для оптимізації нашого проєкту.

## **2 ВИБІР МЕТОДУ УПРАВЛІННЯ ЗА ДОПОМОГОЮ МАІ**

У попередньому розділі ми розглянули методи управління проектами.

У цьому розділі ми визначимо, який з цих методів найкраще підійде у нашому випадку. Так як ми розглянули 4 методи і оцінюємо їх за чотирма характеристиками, то найкраще нам підійде метод аналізу ієрархій.

### **2.1 Огляд методу аналізу ієрархій**

Компанії та інші організації часто стикаються з рішенням вибрати правильні проекти для свого бізнесу. Рішення щодо відбору та реалізації проектів має бути ретельно продумано. Організації стикаються з багатьма різними проблемами та можливостями, які їх оточують. Ми повинні розпізнати правильні можливості та вибрати правильні проекти. Це непросте рішення, і неправильні рішення можуть мати довгострокові згубні наслідки. Ми витрачаємо багато грошей на проекти, які не відповідають потребам і запитам клієнтів. Чи має прийматися рішення суворо на фінансовому аналізі, чи слід враховувати інші критерії [1]

Існує багато різних методів оцінки та відбору проектів. Деякі з них суворо якісні, інші – кількісні. Кожен з цих методів має свої сильні та слабкі сторони. Ми збираємося зосередитися на методі, який називається процесом аналітичної ієрархії (МАІ). Ми розглянемо його основні характеристики, сильні та слабкі сторони та як використовувати його як інструмент для оцінки та відбору проектів. Ми підтримуватимемо використання МАІ для відбору проектів із комп'ютерним додатком у Microsoft Excel. Ми збираємося моделювати різні сценарії важливості критеріїв для конкретного проекту.

Використання МАІ для оцінки та відбору проєктів досліджувалося багатьма авторами [2-4]. Їхні підходи дещо відрізняються, але всі вони корисні. МАІ базується на парних порівняннях з використанням шкали відношень для визначення сили переваги. Мелоун і Уортон [5] обговорювали моделі підрахунку балів у своєму огляді стратегій вибору проєкту та запропонували, що вага має бути присвоєно вартості та аспектам ризику альтернатив, поряд з іншими атрибутами. Шовал і Лугасі [6] розглянули різні підходи для присвоєння ваги вартісному аспекту та для поєднання вартісних і немонетарних оцінок, щоб отримати єдине числове значення для альтернативного вибору. Ми представимо використання МАІ для оцінюємо та відбираємо проєкти на основі їхніх висновків, але в той же час ми відкоригуємо методологію, щоб спростити її використання.

### **2.1.1 Відбір проєкту за стратегічними цілями компанії**

Компанії та інші організації стикаються з новими бізнесовими викликами та проблемами. Їм доводиться справлятися з постійними змінами у своєму бізнес-середовищі та розвиватися. Сьогодні проєкти є засобом реагування на зміни та використання можливостей бізнесу. Також добре відомий факт, що стратегії реалізуються за допомогою проєктів. Стратегічне та проєктне управління все більше взаємопов'язані. Проєкти є ключовими блоками стратегій, оскільки вони перетворюють стратегії в дії [1, 7].

Компанії та організації повинні підготувати багато різних сценаріїв для досягнення необхідних цілей. Ці сценарії, в основному, є проєктами, і ми повинні вибрати ті, які пізніше будуть реалізовані на практиці. Існує багато обмежень, коли ми обираємо проєкти, і ми хочемо вибрати ті, які заощаджують гроші та мають найвищі виплати. Ми повинні підготувати моделі відбору, які допоможуть

нам вибрати правильні проєкти. Соудер і Шерман [8] визначили п'ять важливих питань, які менеджери повинні враховувати при оцінці моделей скринінгу:

- Реалістичність – модель прийняття рішень повинна відображати організаційні цілі та місію.
- Можливості – модель повинна дозволяти компанії порівнювати різні типи проєктів, тобто вона повинна бути достатньо надійною, щоб задовольнити нові критерії та обмеження.
- Гнучкість – модель має бути легко змінена, якщо потрібні зміни. Він повинен дозволяти коригування через зміни валютних курсів, податкового законодавства тощо.
- Простота використання – модель повинна бути достатньо простою, щоб її могли використовувати люди в усіх сферах організації.
- Вартість – модель повинна бути економічно ефективною і не займати багато часу [1].

Як уже зазначалося, моделі відбору проєктів поділяються на два загальні класи: кількісні та якісні [9]. Кількісні моделі використовують числа як вхідні дані для процесу прийняття рішень. Ці значення можуть бути об'єктивними зовнішніми значеннями (кількість цементу, необхідного для будівництва нової дороги) або суб'єктивними внутрішніми цінностями (щоб розробити нову домашню сторінку, нам знадобляться три програміста на шість тижнів). Більшість процесів відбору передбачає поєднання суб'єктивної та об'єктивної оцінки даних та прийняття рішень.

Пінто [1] пропонує чотири групи факторів, які впливають на процес відбору проєкту:

- фактори ризику (технічні, фінансові, безпеку, якість, юридичний вплив);

- комерційні фактори (очікуваний прибуток від інвестицій, термін окупності, потенційна частка ринку, здатність генерувати майбутній бізнес/нові ринки тощо);
- внутрішні діючі фактори (потреба в навчанні працівників, зміна чисельності робочої сили, зміна фізичного середовища, зміна виробничого процесу);
- додаткові фактори (права інтелектуальної власності, вплив на імідж компанії, стратегічна відповідність).

У багатьох ситуаціях існують відповідні атрибути, які неможливо виміряти в грошовому виразі. Вони можуть включати аспекти якості продукції, безпеки та продуктивності; надійність і досвід постачальників; зрілість і стабільність технології, аспекти ризику та невизначеності, а також інші характеристики, які впливають на витрати та вигоди для зацікавлених сторін проекту. Звичайна література щодо відбору проектів пропонує два основних підходи до цієї проблеми [10]. Один підхід полягає у встановленні явних порогових значень для кожного з атрибутів. Далі розглядаються лише альтернативи, які відповідають усім пороговим умовам, і з них вибирається найбільш економічно приваблива альтернатива. Основним недоліком цього підходу є те, що він не розрізняє різні рівні продуктивності за межами порогового значення. Другий і найбільш часто використовуваний підхід полягає у присвоєнні ваги кожному атрибуту, присвоєння числової оцінки кожній з альтернатив щодо кожного з атрибутів та обчисленні суми зважених оцінок для кожної альтернативи. Остаточний вибір ґрунтується на комбінації зваженої суми балів і чистої поточної вартості [11].

### 2.1.2 Процес аналітичної ієрархії

Одним з найкорисніших методів вибору проекту, який стає все більш важливим, є МАІ. Цей метод був розроблений доктором Томасом Сааті у 1980 році як інструмент для допомоги у вирішенні технічних та управлінських проблем [12]. Він спрямований на кількісне визначення відносних пріоритетів для даного набору альтернатив за шкалою співвідношень на основі суджень особи, яка приймає рішення, і підкреслює важливість інтуїтивних суджень особи, яка приймає рішення, а також послідовність порівняння альтернатив. в процесі прийняття рішень.

Оскільки особа, яка приймає рішення, ґрунтує свої судження на знаннях та досвіді, а потім приймає відповідні рішення, підхід МАІ добре узгоджується з поведінкою особи, яка приймає рішення. Сила цього підходу в тому, що він систематично організовує матеріальні та нематеріальні фактори та забезпечує структуроване, але відносно просте рішення проблем прийняття рішень [13]. Крім того, логічно розкладаючи проблему від великого, поступово спускаючись до меншого і меншого, можна за допомогою простих парних порівняльних суджень з'єднати мале з великим.

Метод МАІ дуже підходить для тих сфер, де можна знайти інтуїцію, раціональність та ірраціональність у зв'язку з ризиком та невизначеністю. Проблема може включати соціальні, політичні, економічні та технічні важливі суми, декілька цілей, критеріїв та можливостей. Він використовується для визначення пріоритету (у нашому випадку для важливості факторів і для визначення рівня невизначеності) і для прийняття відповідних рішень. Він розбирає складні проблеми до рівня парних порівнянь, а потім знову об'єднує результати, що веде до раціонально найкращого рішення. На сьогоднішній день

метод МАІ залишається найбільш поширеною і використовуваною теорією прийняття рішень [14].

МАІ особливо підходить для оцінки складних багатопараметричних можливостей із включенням суб'єктивних критеріїв. Ключові етапи застосування методу МАІ охоплюють:

- аналіз загальної проблеми прийняття рішень в ієрархічному сенсі та поділ на підпроблеми, які легше оцінити та зрозуміти,
- призначати пріоритети елементам на кожному рівні ієрархії прийняття рішень,
- призначати числові значення елементам,
- аналіз та оцінка можливих рішень проблеми.

Методологія зваженої оцінки також має деякі недоліки. По-перше, це вимагає від особи, яка приймає рішення, вибрати відповідні значення для ваг атрибута та для оцінок альтернативи. Це може вимагати значних зусиль для досягнення консенсусу між зацікавленими сторонами, які беруть участь у процесі прийняття рішень. Більше того, ваги та бали передбачають певні компроміси між різними рівнями продуктивності на різних альтернативах. Незалежно від методу, який використовується для отримання ваг і балів, ці компроміси є між парами альтернатив за будь-яким даним атрибутом і припускають незалежність від оцінок альтернатив за іншими атрибутами. Отже, вони не враховують взаємозв'язки між різними атрибутами.

### **2.1.3 Використання МАІ для оцінки та відбору проєкту**

Практичне використання кроків МАІ буде продемонстровано пізніше на конкретному проєкті. Але спочатку ми розглянемо конкретні етапи процесу МАІ. Кожен проєкт має свою мету і свої цілі. Вони представляють потреби та бажання



клієнтів. Зазвичай існує кілька можливостей для задоволення цих потреб і бажань [15]. Тому ми готуємо кілька сценаріїв у вигляді проєктів. Для оцінки та відбору проєктів нам потрібні критерії. Замовник проєкту разом із проєктною командою вирішує, які критерії є найкращими для оцінки проєктів. Ми вже представили потенційні критерії у другому розділі.

Першим кроком методу є розробка ієрархії критеріїв. Найвищий рівень - це мета прийняття рішення або мета проєкту. Структурування критеріїв проєкту означає побудову ієрархії критеріїв та її підкритеріїв. Структурування критеріїв у підкритерії допомагає менеджеру встановлювати пріоритети між проєктами. Ієрархія критеріїв відображає структуру організаційної стратегії та ключові показники ефективності та водночас дає можливість вибору проєкту відповідно до його узгодження з бізнес-цілями. Перший виклик, коли ми вибираємо один із стратегічно важливих для нашої організації проєктів, – це встановити відповідні та чіткі критерії. Це також завдання функціональних менеджерів з маркетингу, фінансів, ІКТ, продажів та інших.

Другий крок складається з розподілу ваг для попередньо обраних критеріїв і, за необхідності, розподілу загальної ваги критерію між підкритеріями. Міан і Дай [16] рекомендують так званий підхід попарного порівняння до зважування, в якому кожен критерій порівнюється з кожним іншим критерієм. Ми виконуємо це попарне порівняння на кожному ієрархічному рівні (порівняння двох елементів, які належать до однієї групи всередині ієрархії) і для кожного рівня всієї ієрархії. Таке порівняння дозволяє нам завжди зосереджуватися лише на двох критеріях. Таким чином ми можемо встановити для кожної комбінації, які критерії важливіші, а які менш важливі та яка різниця між ними за важливістю.

Як ми присвоюємо ваги критеріям? Зазвичай ми порівнюємо два критерії одночасно і використовуємо бали від 1 до 9. Обмеження шкали є наслідком

усвідомлення того, що людський розум може правильно відчувати й розглядати лише кілька елементів одночасно. Найточніші рекомендації щодо оцінки пар можна знайти в таблиці I. Кожній парі ми визначаємо ступінь домінування одного елемента над іншим. Виняткову перевагу одного критерію над іншим можна оцінити як 9, рівність 1. Якщо другий критерій важливіший за перший, запишіть взаємне значення. Таким чином ми отримуємо значення в області від 1/9 до 9. Ця модель оцінки коефіцієнтів емпірично підтверджена як достатньо точна для більшості задач. Більша різноманітність суджень призведе до зниження симетрії оцінок. Щоб отримати остаточну оцінку, скористайтеся процедурою середньозваженої оцінки. Це можна отримати, помноживши важливість критеріїв і рівень невизначеності.

Таблиця 2.1 – Значення критеріїв

Значення	Опис порівняння
1	Рівність
3	Деяко більша важливість одного критерію над іншим
5	Сильна перевага одного критерію над іншим
7	Дуже сильна перевага одного критерію над іншим (чітко видно на практиці)
9	Абсолютна (найвища можлива) перевага одного критерію над іншим

Мають, маємо  $f$  критеріїв, важливість яких ми хочемо визначити. Критерії порівнюються попарно (значення від 1 до 9), і ці суб'єктивні оцінки записуються в матрицю вимірів  $f \times f$  [12, 18]. Комірки матриці задаються як найближчі цілі апроксимації вагового відношення.

Таблиця 2.2 – Матриця для оцінки важливості критерію

Критерії	1	2	...	$f$
1	1	$a_{12}$	...	$a_{1f}$
2	$a_{21}$	1	...	$a_{2f}$
.	.	.		.
.	.	.		.
$f$	$a_{f1}$	$a_{f2}$	...	1

Для заповнення матриці нам знадобиться  $(f - 1) / 2$  цінності. Оцінювач дає лише значення (комірки) у верхній частині матриці (над діагоналлю). Комірки на діагоналі дорівнюють 1 (один), значення клітинок під діагоналлю взаємні.

Для практичного використання рівняння або матриці в таблиці II достатньо наближеного рішення. Рекомендуються два методи:

- всі клітинки в окремому стовпці матриці  $[A]$  діляться на суму комірок даного стовпця, потім усі ці значення додаються в рядок, а потім сума ділиться на  $f$  (обчислити середнє значення клітинок у рядку); результатом є вектор пріоритету критеріїв  $\{g\}$ ;
- множити  $f$  клітинок в окремому рядку, обчислити  $f$ -го кореня з добутку (середнє геометричне), потім нормалізуємо отриманий вектор (клітинки діляться сумою стовпця).

Третій крок – призначити числові значення розмірам на нашій шкалі оцінки. Зазвичай використовують розміри, наприклад погано, справедливо, добре, дуже добре і відмінно. Ці значення не підходять для всіх ситуацій, тому ми можемо змінити їх у разі потреби. Цим якісним вимірам ми присвоюємо числове значення від 0 до 1. Це повністю наше рішення, яке значення призначається кожному виміру. Наприклад: погано – значення 0,0; справедлива

– вартість 0,1; добре – значення 0,3; дуже добре – значення 0,6; відмінно – значення 1,0.

Різниці між числами на шкалі нерівні (якщо ми використовуємо значення від 1 до 5, різниці між інтервалами не буде). Це надзвичайно корисно, оскільки менеджери можуть вказати точну різницю у важливості між двома вимірами (у нашому випадку різниця між відмінним і дуже хорошим набагато більша, ніж між справедливим і хорошим).

Четвертий крок забезпечує загальну оцінку проєкту. Спочатку ми помножимо на критерії присвоєні ваги та числове значення проєкту. Загальна оцінка проєкту - це сума всіх добутків ваг і числових значень (балів) проєкту:

$$P = W_1K_1 + W_2K_2 + \dots + W_nK_n \quad (2.1)$$

де:  $p$  - загальна оцінка проєкту,  $W$  – призначена вага критеріїв, і  $K$  – числове значення (оцінка) виміру.

Це процес МАІ для оцінки всіх проєктів. Найкращим проєктом вважається той, який отримав найвищу загальну оцінку (максимальне значення, звичайно, 1).

#### **2.1.4 Приклад використання МАІ для вибору кращого проєкту**

Ми збираємося оцінити чотири проєкти (на основі Pinto [1]). Усі вони є стратегічно важливими для організації Х. Вони повинні сприяти збільшенню Ми збираємося оцінити чотири проєкти (на основі Pinto [1]). Усі вони є стратегічно важливими для організації Х. Вони повинні сприяти збільшенню конкурентоспроможність організації на різних рівнях. Замовники проєкту (вище керівництво організації) повинні були вибрати три критерії для оцінки альтернатив проєкту:

- А – фінансові вигоди,
- В – внесок у стратегію організації,

- С – внесок в ІТ-інфраструктуру.

Критерії фінансових вигод зосереджені на відчутних вигодах від проєкту і далі поділяються на довгострокові та короткострокові вигоди. Внесок у стратегію є нематеріальним фактором, розділеним на три підкритерії: збільшення частки ринку для продукту А, утримання наявних клієнтів для продукту Б та покращення управління витратами.

Таблиця 2.3 – Ієрархія вибору критеріїв відбору

Перший рівень	Другий рівень
1. Фінансові вигоди (А)	А1: короткостроковий А2: довгостроковий
2. Внесок у стратегію організації (В)	В1: збільшення частки ринку для продукту А В2: утримання наявних клієнтів для продукту В В3: покращення управління витратами
3. Внесок в ІТ-інфраструктуру (С)	С1

Спочатку ми повинні порівняти критерії на першому рівні.

Таблиця 2.4 – Вагові значення критеріїв на першому рівні.

	А	Б	С		
А	1	3	5	19000	63,33 %
Б	1/5	1	3	0,7815	26,05 %
С	1/3	1/3	1	0,3185	10,62 %
	1,5333	4,3333	9000	30000	100,00%

Якщо ми подивимося на наш приклад, то побачимо, що критерій фінансової вигоди дещо важливіший, ніж критерій внеску в стратегію організації. При цьому критерії фінансової вигоди набагато важливіші, ніж критерій внеску в ІТ-інфраструктуру. Внесок критерій стратегії організації є дещо важливішим, ніж внесок у критерій ІТ-інфраструктури. Останній стовпець таблиці представляє важливість критеріїв, виражених у відсотках на першому рівні ієрархії.

Два критерії мають додаткові підкритерії. Критерій фінансових вигод має два підкритерії. Для призначення ваг критеріям у цьому випадку ми не використовуємо процес МАІ. Ми просто порівнюємо обидва критерії та визначаємо їх взаємозалежність. Наприклад, ми вважаємо, що довгострокові фінансові вигоди важливіші за короткострокові фінансові вигоди. Ми присвоюємо значення 25 % короткостроковим фінансовим вигодам і 75 % довгостроковим фінансовим вигодам (разом, звичайно, 100 %). Обидва значення мають бути переведені на першому рівні за допомогою такої простої процедури:

$$A_1 = 0.6333 * 0.25 = 0.1583 = 15.83\% \quad (2.2)$$

$$A_2 = 0.6333 * 0.75 = 0.4743 = 47.49\% \quad (2.3)$$

Внесок у критерії стратегії організації має три підкритерії, які ми повинні порівняти. Наш аналіз дав такі значення для всіх з них: збільшення частки ринку для продукту А – 35 %, утримання наявних клієнтів для продукту В – 47,78 % та покращення управління витратами – 17,22 %. Ці значення мають бути перетворені на першому рівні:

$$B_1 = 0,2605 * 0,35 = 0,0912 = 9,12 \% \quad (2.4)$$

$$B_2 = 0,2605 * 0,4749 = 0,1237 = 12,37 \% \quad (2.5)$$

$$B_3 = 0,2605 * 0,1722 = 0,0448 = 4,48 \% \quad (2.6)$$

Тепер у нас є шість критеріїв для оцінки наших чотирьох проектів. Безумовно, найважливішим критерієм є довгострокові фінансові вигоди (47,49

%). Наступним кроком є вибір якісних вимірів і присвоєння їм числових значень за шкалою оцінок. Ми вирішили наступне:

- погано – значення 0,0;
- справедлива – вартість 0,15;
- добре – вартість 0,35;
- дуже добре – значення 0,7;
- відмінно – значення 1,0.

На останньому кроці ми оцінюємо всі чотири проєкти.

Таблиця 2.5 – Оцінка проєкту щодо виконання критеріїв.

		Всього	Фінанси		Стратегія			ІС
			короткострокові	довгострокові	ринок	користувачі	витрати	
			0.1583	0.4749	0.0912	0.1237	0.0448	
1	Проєкт 1		відмінно	відмінно	дуже добре	відмінно	дуже добре	відмінно
2	Проєкт 2		добре	відмінно	добре	відмінно	добре	відмінно
3	Проєкт 3		відмінно	добре	відмінно	добре	відмінно	добре
4	Проєкт 4		дуже добре	дуже добре	дуже добре	дуже добре	дуже добре	дуже добре

Ми повинні проаналізувати кожен проєкт окремо і визначити, який рівень виконання кожного обраного критерію. Значення «відмінно» означає, що даний

проект повністю відповідає певному критерію. Значення «добре» означає, що цей проект відповідає критерію набагато менше, ніж повинен.

Далі ми перетворюємо якісні значення з таблиці на числові.

Таблиця 2.6 – Розрахунок загальної оцінки проекту.

		Всього	Фінанси		Стратегія			ІС
			короткострокові	довгострокові	ринок	користувачі	витрати	
			0.1583	0.4749	0.0912	0.1237	0.0448	
1	Проект 1		1.00	1.00	0.70	1.00	0.70	1.00
2	Проект 2		0.35	1.00	0.35	1.00	0.35	1.00
3	Проект 3		1.00	0.35	1.00	0.35	1.00	0.35
4	Проект 4		0.70	0.70	0.70	0.70	0.70	0.70

Розрахунок загальної оцінки для першого проекту:

$$P_1 = 0,1583 \cdot 1,00 + 0,4749 \cdot 1,00 + 0,0912 \cdot 0,70 + 0,1237 \cdot 1,00 + 0,0448 \cdot 0,70 + 0,1062 \cdot 1,00 \quad (2.7)$$

$$P_1 = 0,9583 \quad (2.8)$$

Загальні оцінки для інших трьох проектів наведені в таблиці VI. Ми бачимо, що проект 1 отримав найвищу оцінку. Очевидно, це найкращий вибір для нашої організації. Цікаво проаналізувати Проект 2 і Проект 3. Вони обидва



отримали однакову кількість значень «відмінно» та «добре». Але різниця в загальній оцінці проєкту вкрай різна. Причиною цього є той факт, що Проєкт 2 отримав чудову цінність за найбільш домінуючим критерієм – довгостроковою фінансовою вигодою.

### **2.1.5 Програмне забезпечення MS EXCEL для процесу МАІ**

Існує багато різного програмного забезпечення, яке допомагає використовувати процес МАІ, але вони в основному комерційні та досить дорогі (наприклад, Expert Choice). Тому ми розробили просту комп'ютерну програму в широко популярному Microsoft Excel 2007. Ця програма надає допомогу з простими процедурами МАІ. Для використання програми потрібно виконати кілька кроків:

- Спочатку ми повинні вирішити, скільки критеріїв ми будемо використовувати. На даний момент ця програма дозволяє використовувати 10 критеріїв. Інше обмеження полягає в тому, що всі критерії мають бути на першому рівні. Як це робиться, ми продемонстрували на описаному вище прикладі.

- Додаток має матрицю  $10 \times 10$ , куди ми вводимо суб'єктивні оцінки порівнюваних пар критеріїв. Ми використовуємо тільки верхню частину матриці (зверху справа від діагоналі). Значення під діагоналлю обчислюються автоматично. Коли ми вводимо значення суб'єктивної оцінки, ми повинні враховувати, який критерій є першим, а який – другим. Наприклад; якщо критерій А набагато важливіший за критерій В, то вводимо значення 7. Якщо критерій В набагато важливіший за критерій А, то вводимо значення  $1/7$  (рис. 1).

	A	B	C	D	E	F	G	H	I	J										
1	The numer of criteria				4															
4	A	B	C	D	E	F	G	H	I	J										
5	1,0000	3,00	5,00	3,00																
6	0,3333	1,0000	2,00	3,00																
7	0,2000	0,5000	1,0000	2,00																
8	0,3333	0,3333	0,5000	1,0000																
9	0,0000	0,0000	0,0000	0,0000	1,0000															
10	0,0000	0,0000	0,0000	0,0000	0,0000	1,0000														
11	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	1,0000													
12	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	1,00												
13	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	1,0000										
14	1,8667	4,8333	8,5000	9,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000										
17	Matrix																			
19	A	B	C	D	E	F	G	H	I	J										
20	0,5357	0,6207	0,5882	0,3333	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	2,0780	51,95%								
21	0,1786	0,2069	0,2353	0,3333	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,9541	23,85%								
22	0,1071	0,1034	0,1176	0,2222	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,5505	13,76%								
23	0,1786	0,0690	0,0588	0,1111	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,4175	10,44%								
24	0,0000	0,0000	0,0000	0,0000	1,0000	0,0000	0,0000	0,0000	0,0000	0,0000	1,0000	0,00%								
25	0,0000	0,0000	0,0000	0,0000	0,0000	1,0000	0,0000	0,0000	0,0000	0,0000	1,0000	0,00%								
26	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	1,0000	0,0000	0,0000	0,0000	1,0000	0,00%								
27	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	1,0000	0,0000	0,0000	1,0000	0,00%								
28	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	1,0000	1,0000	1,0000	0,00%								
30												100,00%								

Рисунок 2.1 – Розрахунок важливості критеріїв за допомогою програми MAI MS Excel

Додаток автоматично розраховує вагові коефіцієнти важливості критеріїв. Ці значення також автоматично переносяться на наступну фазу. Першим завданням на другому етапі є встановлення шкали оцінок для якісних вимірів проєкту.

Кожен якісний вимір отримує числове значення, і ці значення заносяться до другої матриці. Додаток дозволяє аналізувати 6 проєктів одночасно. Програма розраховує загальну оцінку для всіх проєктів на основі ваг критеріїв і якісних розмірів, виражених у числових значеннях .

### 2.1.6 Моделювання критеріїв

Додаток MS Excel не тільки дозволяє просто обчислити загальну оцінку проєкту на основі процесу MAI, але також допомагає в процесі прийняття рішень,

коли нам доводиться моделювати різні ситуації. Ці ситуації стосуються зміни важливості встановлення критеріїв.

Criteria	A	B	C	D	E	F	G	H	I	J	Total grade
Weight	0,5195	0,2385	0,1376	0,1044	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	
Project 1	0,2597	0,2385	0,1376	0,0522	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,6881
Project 2	0,5195	0,1789	0,1032	0,0261	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,8277
Project 3	0,3896	0,0596	0,1376	0,1044	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,6912
Project 4	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Project 5	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Project 6	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000

Рисунок 2.2 – Розрахунок загальної оцінки проекту за допомогою програми MAI MS Excel

Процес визначення відмінностей у важливості двох критеріїв все ще є дуже суб'єктивною оцінкою. Змінюючи значення в парних порівняннях критеріїв, ми можемо відразу побачити вплив цих змін на загальну оцінку проекту. Ми збираємося представити наслідки цих змін на гіпотетичному прикладі з двома проектами, які оцінюються за 5 критеріями. Припустимо, що початкове порівняння критеріїв виконано з результатами, представленими в таблиці VII.

Ми бачимо, що критерій A є найважливішим (48,99 %). Ми також бачимо, що на основі якісної оцінки обох проектів з огляду на виконання критеріїв, проект 1 є дещо кращим вибором, ніж проект 2.

Немає жодних суперечок щодо того, наскільки обидва проекти відповідають обраним критеріям – усі менеджери повністю згодні. Але менеджери не погоджуються щодо важливості конкретних критеріїв. Топ-менеджер компанії вважає, що критерій А ще важливіший у порівнянні з іншими чотирма критеріями. Він хоче підвищити значення з 3 до 5, з 5 до 7 і з 7 до 9. Що відбувається? Критерій А отримує вагу 57,72 %. Але як це впливає на загальну оцінку проекту? Аналіз показує, що проект 1 не має найвищої загальної оцінки. Раптом Проект 2 отримує вищу оцінку, і це кращий вибір, ніж Проект 1 (Проект 1: 0,7671 і Проект 2: 0,7974). Симулюючи відмінності у важливості критеріїв, ми можемо миттєво побачити, що відбувається із загальною оцінкою проектів. Це прямий доказ того, що вибір критеріїв та визначення їх важливості має бути дуже ретельно реалізованим процесом.

Таблиця 2.7 – Приклад 1

	А	Б	С	Д	Е	Вага
А	10000	3,00	5,00	5,00	7,00	48,89 %
Б	0,3333	10000	3,00	1,00	5,00	20,57 %
С	0,2000	0,3333	10000	3,00	3,00	14,37 %
Д	0,2000	10000	0,3333	10000	3,00	14,41 %
Е	0,1429	0,2000	0,3333	0,3333	10000	4,63 %

Таблиця 2.8 – Продовження прикладу 1

Критерій \ Проект	А	В	С	Д	Е	Ф
Проект 1	0.75	0.4	0.4	0.75	0.75	0,7728
Проект 2	1	0.4	0.4	0.75	0.4	0,7553

Інша ситуація може виникнути, коли ми оцінюємо та відбираємо проекти. Цілком можливо, що нам доведеться змінити певні характеристики проекту.

Можливо, ми вирішили зменшити або розширити обсяг проєкту, змінити конкретні цілі, усунути, змінити чи додати деякі ресурси, видалити чи додати деякі дії тощо. У цьому випадку є велика ймовірність того, що проєкт раптом по-іншому відповідає обраним критеріям. Тому ми повинні знову оцінити кожен проєкт і визначити рівень відповідності кожному критерію.

Ми будемо використовувати вищеописаний випадок. Критерієм В у нашому випадку є рівень наявних коштів на проєкт. Нам довелося додати до Проєкту 1 певні людські ресурси, щоб покращити якість кінцевих результатів проєкту. Додаткові ресурси були зовнішні і набагато дорожчі за внутрішні людські ресурси. Тому довелося підвищити бюджет проєкту. Раптом наш Проєкт 1 відповідає критерію В на набагато нижчому рівні – наприклад, рівень добре. Подивіться, що відбувається. Проєкт 1 отримує набагато нижчу загальну оцінку.

Таблиця 2.8 – Приклад 2

Критерії \ Проєкт	A	B	C	D	E	F
Проєкт 1	0.75	1	0.4	0.75	0.75	0,6493
Проєкт 2	1	0.4	0.4	0.75	0.4	0,7553

Програмний додаток допомагає підрахувати загальні оцінки проєкту, коли ми моделюємо різні рівні виконання критеріїв.

Нині процес МАІ використовується в різних ситуаціях прийняття рішень. Ми вирішили представити його використання для оцінки та відбору проєктів. МАІ може значно покращити процес розробки проєктних пропозицій. Його найбільшою перевагою є системний підхід у кілька етапів та його здатність знижувати суб'єктивність менеджерів, яким доводиться вибирати між альтернативами проєкту. З іншого боку, МАІ має кілька недоліків при

використанні для відбору проєктів. Перший недолік полягає в тому, що він ігнорує, що певний вибір може призвести до негативних результатів. Друге обмеження полягає в тому, що МАІ вимагає, щоб усі критерії були повністю розкриті та враховані початок процесу відбору. Це також дозволяє більш потужним членам організації вболівати за власні проєкти та перешкоджати відкритому процесу відбору. Сам процес може бути досить складним для розуміння, він також вимагає певних математичних зусиль. Тому ми розробили простий у використанні програмний додаток, який може допомогти менеджерам при оцінці проєктних пропозицій. У той же час цей інструмент забезпечує просте моделювання.

## **2.2 Вибір кращого методу управління використовуючи МАІ**

У цьому підрозділі ми виберемо метод управління проєктами за допомогою методу аналізу ієрархій.

Характеристики за якими ми будемо оцінювати системи управління ми формуємо у вигляді критеріїв.

Критерій А. Пріоритетність задач

Критерій В. Ретроспектива

Критерій С. Командна оцінка задач

Критерій D. Визначені терміни проєкту

Основна мета – вибрати метод який би найбільше підходив за всіма критеріями одночасно (якщо це можливо).

Як альтернативи розглядалися чотири методи, які були описані у Розділі 1, а саме: класичний проєктний менеджмент, scrum, kanban та six sigma.

Таблиця 2.9 Характеристика методів управління проектами

	класичний проектний менеджмент	scrum	kanban	six sigma
Пріоритетність задач		+	+	+
Ретроспектива		+		
Командна оцінка задач		+	+	
Визначені терміни проєкту	+			+

Далі ми будемо ієрхачну модель «ціль – критерії – альтернативи».

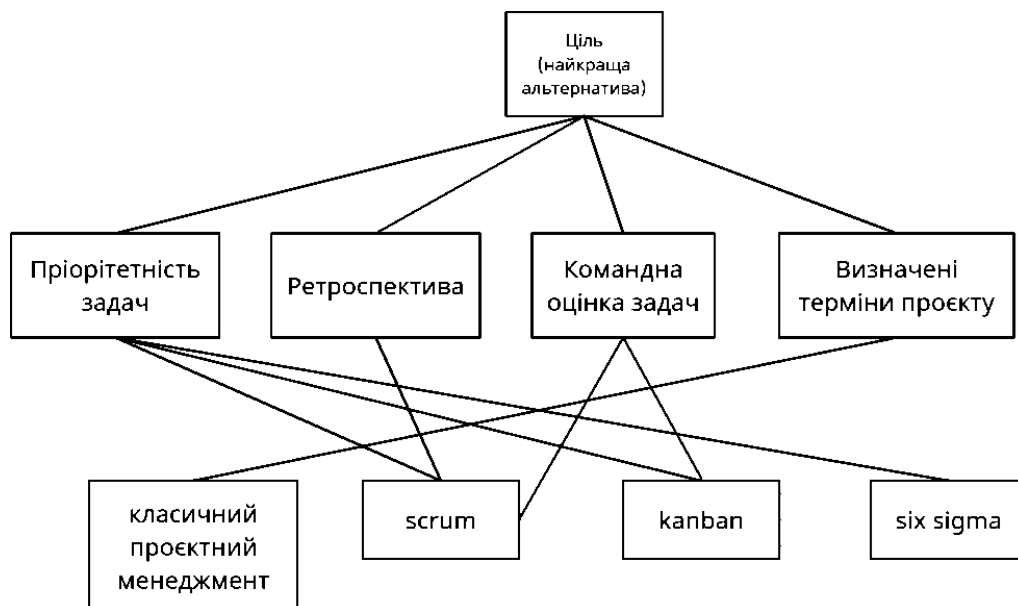


Рисунок 2.5 – Ієрархічна модель

Далі ми виконуємо експертне порівняння переваг кожного з критеріїв над усіма іншими, використовуючи терміни шкали Сааті.

Критерії 2 та 4 – не переважають жоден з інших критеріїв. Критерії 3 – слабо переважають критерії 2 та 5. Критерій 1 – переважає критерії 3, і суттєво переважає критерії 2,4.

Заповнюємо матрицю попарних порівнянь критеріїв за шкалою відносної важливості Сааті. При заповненні матриці та виконанні наступних обчислень ми використовуємо застосунок MS Excel

	A	B	C	D
A	1	5	6	7
B	0,20	1	4	6
C	0,17	0,25	1	4
D	0,14	0,17	0,25	1

Рисунок 2.6 – Матриця попарних порівнянь критеріїв

Обчислити компоненти головного власного вектора матриці та компоненти вектора пріоритетів.

$$V_i = \sqrt[n]{\prod_{j=1}^n a_{ij}}, \quad P_i = \frac{V_i}{\sum_{i=1}^n V_i}, \quad i=1, n \quad (2.9)$$

$$\lambda_i = \frac{\sum_{j=1}^n a_{ij} V_j}{V_i} \quad (2.10)$$

Обчислити коефіцієнти  $\lambda_i$  за формулою (2.10) та записати у додатковий стовпець. Знайти максимальне власне значення матриці, яке є усередненою мірою узгодженості матриці:



				sum	sum/count
0,66	0,78	0,53	0,39	2,36	0,591
0,13	0,16	0,36	0,33	0,98	0,244
0,11	0,04	0,09	0,22	0,46	0,115
0,09	0,03	0,02	0,06	0,20	0,050
					1

Рисунок 2.7 – Максимальне власне значення матриці, яке є усередненою мірою узгодженості матриці.

Далі ми знаходимо індекс узгодженості, оцінку не узгодженості та відношення узгодженості.

$$I_0 = \frac{I_u}{I_{random}}, \quad I_u = \frac{\lambda_{\max} - n}{n - 1}, \quad (2.11)$$

$$I_{random} = 1,12 \quad (2.11)$$

	Добуток матриці на вектор		
	0,59	2,85	4,82
	0,24	1,12	4,59
	0,12	0,47	4,11
	0,05	0,20	4,10
			4,41
Індекс узгодженості ІУ (ІС)			0,14

Рисунок 2.7 – Знаходження індексу узгодженості

	Оцінка неузгодженості			
	A	B	C	D
A	1	5	6	7
B	0,20	1	4	6
C	0,17	0,25	1	4
D	0,14	0,17	0,25	1

Рисунок 2.8 – Знаходження оцінки неузгодженості

Відношення узгодженості (УВ ОС)	IУ/ВІ	0,15

Рисунок 2.9 – Знаходження відношення узгодженості

За результатами наших розрахунків при відношенні узгодженості 0,15, найкращий показник у нас має альтернатива А. Отже найкраще нам підходить класичний проєктний менеджмент, який ми будемо застосовувати в розробці нашої оптимізації для кавярні.

### 3 РОЗРОБКА СИСТЕМИ ДЛЯ ОПТИМІЗАЦІЇ РОБОТИ КАВ'ЯРНІ

У цьому розділі ми описуємо розроблений веб застосунок для оптимізації роботи кав'ярні. Суть застосунку використати класичний проєктний менеджмент для оптимізації роботи.

Для даного веб застосунку було розроблено базу даних та створено таблиці, що відображають основні сутності програми.

Було додано сутність користувач (таблиця users). Дана таблиця містить наступні поля (колонки): id (індифікатор користувача), name (назву користувача), password (пароль для авторизації) та created\_at (дату створення). Дані з поля name будуть відображатись при авторизації на сторінці login та на початковій панелі адміністрування. Також імена користувачів знадобляться нам при формуванні статистики. Пароль користувача буде використовуватись для аутентифікації. Для цього процесу також було розроблено окрему таблицю та програмну сутність: authorization.

Дана таблиця містить поля: id (індифікатор авторизації), user\_id(індифікатор користувача), auth\_token (ключ авторизації), created\_at (дата створення). Дана таблиця є відображенням сутності авторизації. Після вводу авторизаційних даних (паролю) користувачем програма варифікує їх, порівнюючи з даними з таблиці, після чого створюється новий запис в таблиці authorization, який містить зв'язок з користувачем через поле user\_id та згенерований auth\_token та дату створення запису, що є фактичною датою авторизації користувача. Ключ авторизації auth\_token програма вертає у відповіді після запиту на сервер та додає його в куки запиту. Даний ключ буде використовуватись для подальшої авторизації користувача в системі.

```
<div class="auth_box">  
  
<div id="changeform" style="position: relative;">  
  <br>  
  <button type="button" id="registernow1" class="auth_btn">Реєстрація</button><br>  
  <button type="button" id="vvoiti" class="auth_btn btn_green">увійти</button><br>  
</div>  
</div>
```

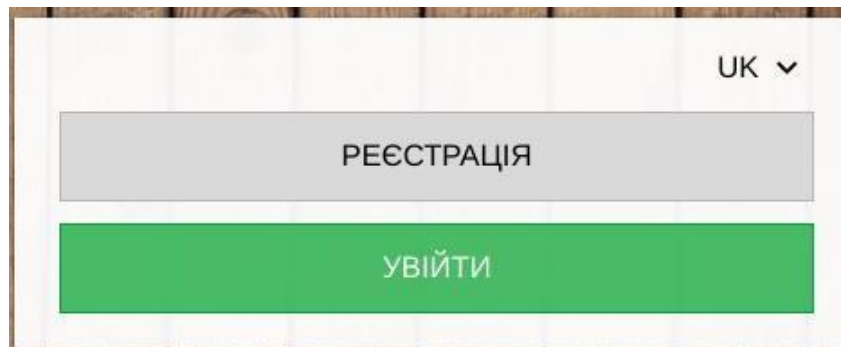


Рисунок 3.1 – Початкова форма входу та реєстрації

Продавець має можливість перейти на сторінку вибору користувача та обрати його.

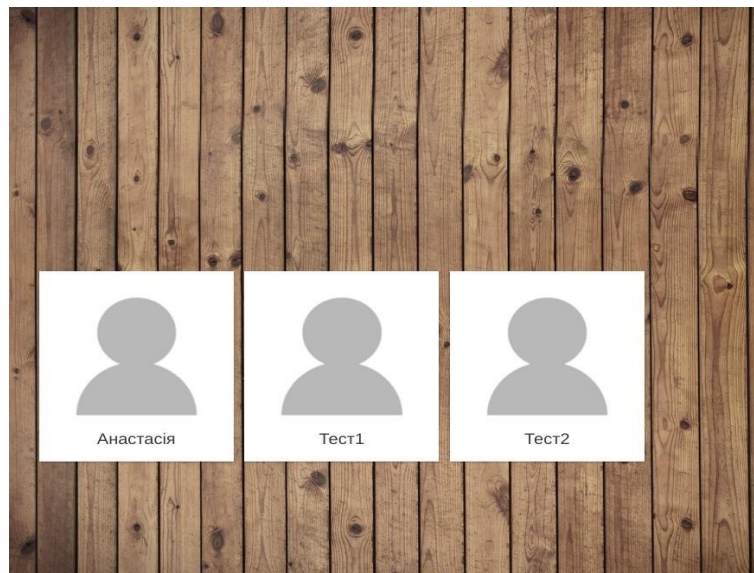


Рисунок 3.2 – Сторінка вибору користувача

Після вибору продавець повинен ввести авторизаційні дані – пароль, після чого він буде переспрямований на сторінку оформлення замовлень.

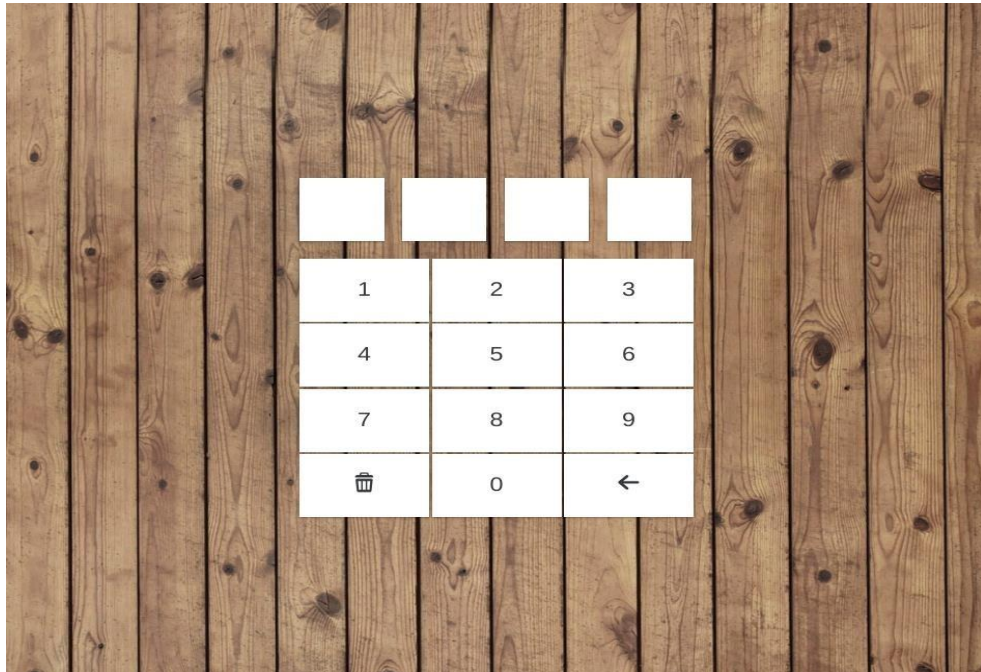


Рисунок 3.3 – Сторінка авторизації користувача

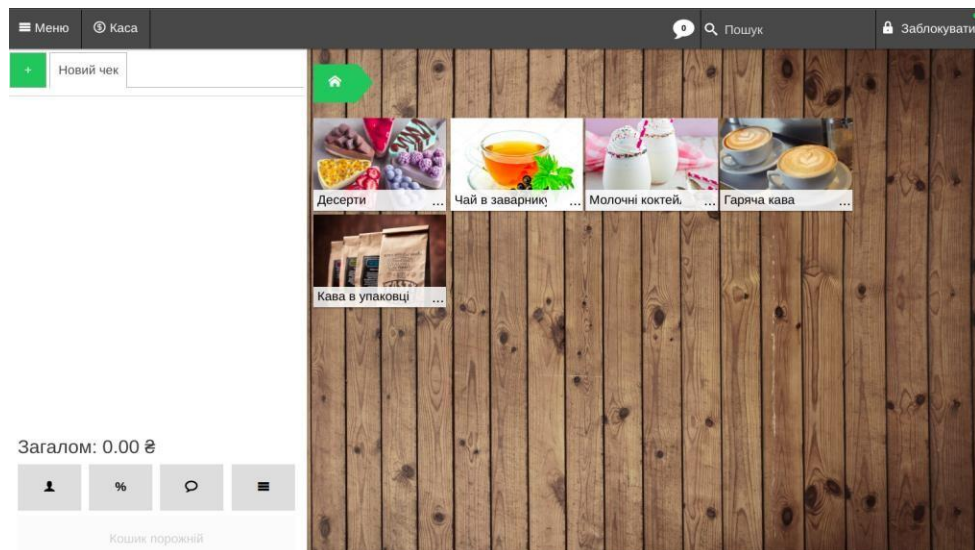


Рисунок 3.4 – Панель оформлення замовлення

Дана сторінка містить верхню навігаційну панель, панель чеку, та панель із категоріями страв. Навігаційна панель допомагатиме користувачу швидко переходити по сторінках веб застосунку. Меню навігаційної панелі містить наступні категорії: Початкова (початкова сторінка панелі адміністрування), Товари (сторінка з категоріями та товарами), Статистика (сторінка, що надає можливість користувачу сформувати статистику), Фінанси, Маркетинг, Налаштування, Допомога та Вихід. Також на панелі навігації користувач може обрати касу. З сторони бекенду каса – це також окрема сутність, яка відображається таблицею `cash_register`. Дана таблиця має поля `id` (індифікатор каси), `name` (назва каси, наприклад: Каса 1), `created_at` – дата створення каси.

При створенні нового чеку користувач має можливість обрати касу та додати страви.

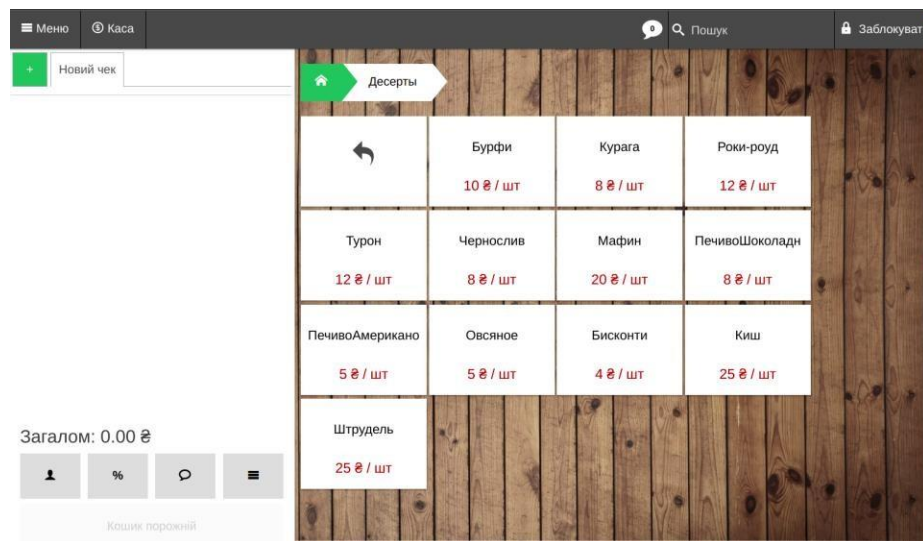
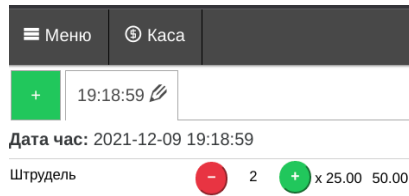


Рисунок 3.5 – Панель вибору страви



Загалом: 50.00 €

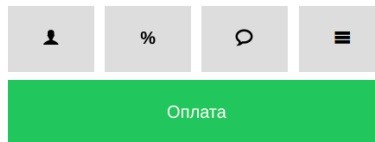


Рисунок 3.6 – Відображення страв в замовленні

Тому таблиця чеку (check) містить поля: id (індифікатор чеку), cash\_register\_id (каса), dish\_id (блюдо), user\_id (продавець), created\_at(дата створення чеку), amount (ціна чеку), status (статус чеку: pending – чек очікує на оплату, payed – чек оплачений). Ціна чеку рахується як сума всіх товарів в чеку, що помножена на їх кількість. Після додавання товарів в чек користувач повинен натиснути – Додати в корзину, після чого створюється чек із статусом pending.

Кожен товар закладу повинен мати свою категорію, тому для цього було створено таблицю categories з полями: id (індифікатор категорії), name(назвою категорії), created\_at (дата створення категорії). Сутність товар item має поля: id(індифікатор товару), name(назва товару), description(опис товару), price(ціна товару), category\_id(категорія товару), created\_at(дата створення товару). При

виборі страви або товару для чеку користувач повинен вибрати категорію, а вже потім – сам товар або страву. Після вибору товару користувач має можливість обрати к-сть товару в чеку, ціна товару з урахуванням його к-сті рахується автоматично. Користувач має можливість додати необмежену к-сть страв або товарів в чек. Після додавання чеку в корзину користувач має можливість натиснути на кнопку Оплата, яка переспрямує користувача на сторінку оплати.

The screenshot displays a payment form with the following elements:

- До виплати:** 50
- Готівка:** 500
- Безгот.:** (empty field)
- Решта:** 450.00
- Клієнт:** (empty field)
- Payment Options:**
  - Без оплати
  - Роздрукувати чек
- Keypad:** A grid of buttons for digits 1-9, 0, a decimal point, and a back arrow.
- Action Buttons:**
  - Скасовувати (Cancel)
  - Оплатити (Pay) - highlighted in green

Рисунок 3.7 – Оформлення оплати товару

На даній сторінці користувач може обрати поля: До оплати, Готівка або Безготівк, Решта та Клієнт. Також є можливість обрати checkbox – без оплати, при виборі цього поля інші поля будуть неактивні. Після проведення оплати користувач буде переспрямований на сторінку звіту або чеку оплати.

Будь який заклад харчування повинен враховувати також операції логістики. В повсякденному житті ми часто зустрічаємось з такими поняттями, як прихід товару, списання, інвентаризація. Для даних операцій було розроблено таблицю `goods_receipts`, що містить поля: `id` (індифікатор приходу товару) `date`(дата приходу товару) `type` (`Charge-off` – списання коштів, `On-credit` – в кредит), `vendor_name` (назва організації постачальника), `comment`(коментар),



items(товари, що надійшли). Для оформлення надходження товару користувачу необхідно заповнити дату надходження, вибрати тип надходження (в борг чи списати кошти) та список товарів. При заповненні списку товарів користувачеві необхідно ввести назву товару, наявність штрихкоду, к-сть, ціну товару за одну одиницю. Автоматично буде розрахована загальна сума для одного товару.

Новий прихід

Дата: 12/09/2021 07:24:45 PM

Категорія витрат: Прихід товару Постачальник: Капітошка

В борг  Списати кошти

Коментар:

Список товарів

Найменування	Штрихкод	К-сть	Ціна	Сума	Ціна роз.
Кекси		шт 20		0 ₪	0 ₪

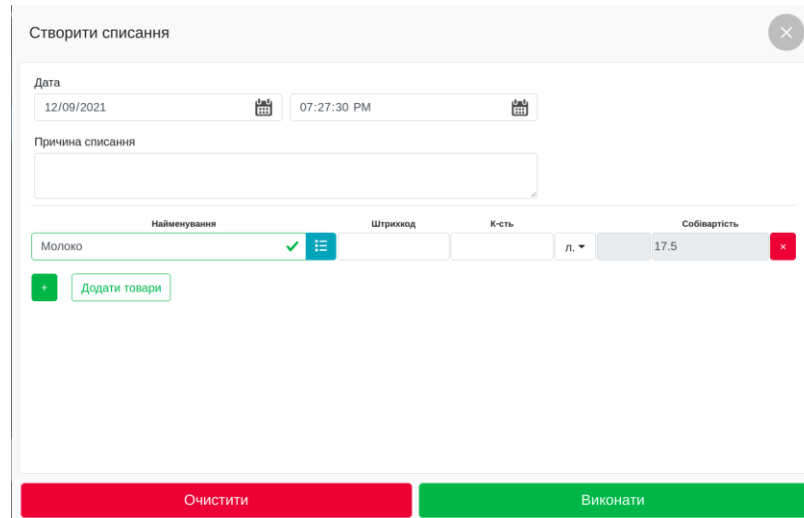
Додати товари

Очистити Виконати

Рисунок 3.8 – Звіт за зміну

Після натискання Виконати, програма створить товари в базі даних та створить запис в таблиці goods\_receipts. В таблиці goods\_receipts було створене спеціальне поле типу json - items, для зберігання масиву з ідентифікаторами товарів в форматі json.

Також варто звернути увагу на списання товарів. Даний веб застосунок надає можливість це зробити. Для цього необхідно перейти в розділ списання товарів, вибрати дату та причину списання, обрати наявні товари та натиснути Виконати.



Створити списання

Дата: 12/09/2021 07:27:30 PM

Причина списання:

Найменування	Штрихкод	К-сть	Собівартість
Молоко		л.	17.5

Додати товари

Очистити Виконати

Рисуно 3.9 – Сторінка списання

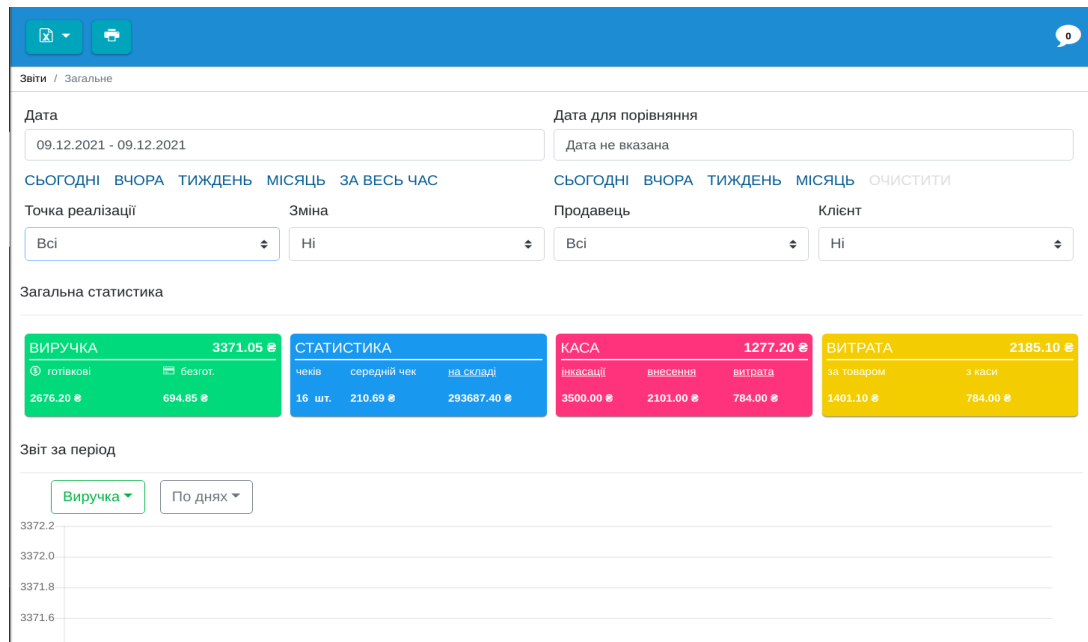


Рисунок 3.10 – Статистика

Рисунок 3.10 показує реалізацію функції системи збору і відображення статистики, яка будується на даних про оплату.

## **4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ**

### **4.1 Організація охорони праці в закладах громадського харчування**

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних та лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я та працездатності людини в процесі трудової діяльності [20]. В даній роботі розглядається управління закладами харчування та здійснюється розробка оптимізованої системи управління закладом харчування. Тому при управлінні закладом харчування необхідним є дотримання правил охорони праці, техніки безпеки та протипожежної безпеки. Під час роботи закладу харчування використовується багато електричних та інших приладів під час використання яких, може статись певна надзвичайна ситуація і потрібен детальний інструктаж з техніки безпеки перед роботою.

Техніка безпеки – це система організаційних заходів і технічних засобів, що запобігають дії на працівників небезпечних факторів [20].

Заклади ресторанного господарства містять у собі цілий комплекс організації послуг харчування, виробництва кулінарної продукції, завдяки використанню різноманітних технологічних машин, що відповідають вимогам естетики та ергономіки, дотримання людиною вимог охорони праці, техніки безпеки. Людина приймає безпосередню участь у виробничому процесі. Людський фактор здійснює сильний вплив на неоднорідність, мінливість вимог охорони праці, техніки безпеки і є важливою проблематикою у сфері розвитку закладів ресторанного господарства. Для вирішення проблематики розробляється комплекс обов'язкових заходів з охорони праці й техніки безпеки.

Організація роботи з охорони праці в закладах ресторанного господарства, як і інших підприємствах, покладається на керівника підприємства або на роботодавця. У відповідності до Закону про охорону праці [19], у невеликих закладах ресторанного господарства наказом призначають працівника, відповідального за організацію роботи з охорони праці. Дотримання правил охорони праці та техніки безпеки має велике значення при експлуатації всіх видів устаткування.

Основними вимогами безпеки устаткування закладів ресторанного господарства є збереження здоров'я та життя людини, надійність та зручність експлуатації.

Безпека устаткування забезпечується:

- вибором безпечного принципу роботи, конструктивних схем, елементів конструкцій;
- використанням засобів механізації, автоматизації та дистанційного керування;
- застосування в конструкціях засобів захисту; – дотримання ергономіки та естетики;
- включення вимог безпеки в технічну документацію з транспортування, монтажу, експлуатації, ремонту, зберігання устаткування. Загальні вимоги безпеки до технологічного устаткування регламентуються нормативними документами, якими передбачається:
  - зміна технологічних процесів та операцій, пов'язаних з виникненням небезпечних та шкідливих факторів процесами та операціями, при виконанні яких ці фактори відсутні або мають меншу дію;
  - механізація й автоматизація технологічних операцій;

- застосування дистанційного керування технологічними процесами та операціями за наявності небезпечних та шкідливих факторів;
- герметизація;
- раціональна організація праці та відпочинку з метою профілактики монотонності та зниження важкості праці;
- своєчасне отримання інформації про виникнення небезпечних та шкідливих виробничих факторів на окремих технологічних операціях;
- запровадження системи керування технологічними процесами, які забезпечать захист працівників та аварійне вимикання устаткування;
- забезпечення пожежної безпеки.

Адміністратор закладу харчування перед зачиненням повинен особисто впевнитись у працездатності автоматичних систем виявлення та гасіння пожеж наприклад Спринклерної водяної системи, оглянути приміщення, переконатися у відсутності порушень, що можуть призвести до пожежі, і тільки після цього останнім залишити приміщення та зачинити двері.

Для безпеки процесів готування їжі та обслуговування споживачів офіціанти та бармени повинні дотримуватися певних вимог охорони праці та техніки безпеки:

- дотримуватися порядку і черговість отримання готових страв з кухні при обслуговуванні споживачів;
- ставити страви на піднос тільки в один ряд;
- не носити столові прилади (ножі , виделки ) у руках вістрям вперед, використовувати для цього тарілку або піднос;
- відкривати пляшки тільки штопором або ключем;
- не заколювати одяг шпильками;
- не тримати в кишенях предмети які б'ються і гострі;

– регулярно перевіряти стан електричних апаратів і механізмів, приділяючи особливу увагу вмикачам, шнурам та розеткам.

Сучасні підходи до охорони праці, техніки безпеки, естетики й ергономіки устаткування в закладах ресторанного господарства спрямовані на збереження здоров'я людини, майна, покращення умов праці, підвищення продуктивності праці, задоволення роботою, випуску якісної кулінарної продукції тощо.

#### **4.2 Шкідливий вплив іонізуючого випромінювання**

Іонізуючі випромінювання знаходять широке використання в різних галузях промисловості. Їх використовують для автоматичного контролю технологічних процесів, контролю якості виробів, зварних швів, структури металів тощо [44].

Для виробництва електроенергії на атомних електростанціях необхідне ядерне паливо, виробництво якого, починаючи від добування уранової руди і закінчуючи виготовленням та транспортуванням паливних елементів, призводить до опромінення персоналу. Незначні додаткові дози опромінення працівники отримують від таких техногенних джерел, як теплові електростанції (підвищена активність їх відходів та аерозолів), підприємств, які пов'язані з видобуванням та переробкою корисних копалин, а також різноманітних приладів та обладнання з джерелами випромінювання, що знаходять широке використання у промисловості і сільськогосподарському виробництві.

Основним документом, що встановлює радіаційно-гігієнічні регламенти для забезпечення прийнятих рівнів опромінення, є Норми радіаційної безпеки України (НРБУ-97).

НРБУ-97 регламентують опромінення людини джерелами іонізуючого випромінювання в умовах:

- нормальної експлуатації індустриальних джерел іонізуючого випромінювання;
- медичної практики;
- радіаційних аварій;
- опромінення техногенно-підсиленими джерелами природного походження.

Відповідно до цього НРБУ-97 встановлено чотири групи радіаційно-гігієнічних регламентів:

- перша – обмежує опромінення від ядерно-радіаційних об'єктів;
- друга – обмежує опромінення людей від медичних джерел;
- третя – обмежує опромінення в умовах радіаційних аварій;
- четверта – обмежує опромінення від техногенно підслених джерел природного походження.

Враховуючи різнобічні наслідки опромінення людей іонізуючим випромінюванням, їх нормування здійснюється залежно від категорії людей, що опромінюються, а також від чутливості органів тіла людини, на які діє іонізуюче випромінювання.

Виділяють наступні категорії:

А – особи з числа персоналу, які постійно чи тимчасово працюють безпосередньо з джерелами іонізуючого випромінювання;

Б – особи з числа персоналу, які безпосередньо не зайняті роботою з джерелами іонізуючого випромінювання, але у зв'язку з розташування робочих місць в приміщеннях та на промислових майданчиках об'єктів з

радіаційноядерними технологіями можуть отримувати додаткове опромінення; В – все населення.

Частину населення, яке за своїми статевовіковими, соціально-професійними умовами, місцем проживання та іншими ознаками може отримувати найбільші рівні опромінення від даного джерела, прийнято виділяти як критичну групу.

Для осіб категорій А і Б НРБУ-97 встановлюються ліміти річних ефективних доз зовнішнього опромінення, а також ліміти річних еквівалентних доз зовнішнього опромінення окремих органів і тканин людини. Аналогічні ліміти вводяться і для критичних груп осіб категорії В. Ліміти дози опромінення наведені в табл. 4.2.

Таблиця 4.2 – Ліміти дози опромінення (мЗв/рік)

Назва лімітів	Категорія осіб, які зазнають опромінення		
	А	Б	В
<i>ЛД<sub>E</sub></i> (ліміт ефективної дози)	20*	2	1
Ліміт еквівалентної дози зовнішнього опромінення:	150		
<i>ЛД<sub>lens</sub></i> (для кришталика ока)	500	15	15
<i>ЛД<sub>skin</sub></i> (для шкіри)	500	50	50
<i>ЛД<sub>extrim</sub></i> (для кистей та стіп)		50	-

Є також обмеження стосовно швидкості накопичення дози для жінок дітородного віку та вагітних жінок, підвищеного опромінення в непередбачуваних ситуаціях та інші.



Крім лімітів дози опромінення, встановлюють допустимі рівні (ДР): потужності дози зовнішнього опромінення, забруднення поверхонь, надходження радіонуклідів через органи дихання тощо, які визначають виходячи із наведених лімітів дози опромінення.

З метою зниження рівнів опромінення населення Міністерство охорони здоров'я України запроваджує рекомендовані рівні медичного опромінення. При проведенні профілактичного обстеження населення річна ефективна доза не повинна перевищувати 1 мЗв.

Медичне опромінення – це опромінення працівників при медичних обстеженнях чи лікуванні. Опромінення повинно бути обґрунтованим і призначеним тільки лікарем для досягнення корисних діагностичних та терапевтичних ефектів, які неможливо отримати іншими методами діагностики та лікування.

Рекомендовані рівні медичного опромінення та детальні вимоги до обмеження і контролю за опроміненням пацієнтів регламентуються окремими спеціальними документами Міністерства охорони здоров'я України. При проведенні профілактичного медичного обстеження працівників річна ефективна доза не повинна перевищувати 1 мЗв.

Для радіометричного і дозиметричного контролю використовуються: дозиметри – для вимірювання зовнішніх потоків радіоактивного випромінювання; радіометри – для вимірювання рівнів забруднення навколишнього середовища; індивідуальні дозиметри – для індивідуального контролю.

Серед індивідуальних дозиметрів найбільше розповсюджені прилади, в яких використовують іонізаційні (за величиною іонізації середовища, через яке

пройшло випромінювання) та фотографічні (за величиною опромінення фотографічної плівки іонізуючим випромінюванням) методи виміру.

У приладах для контролю потужності дози випромінювання широко застосовують іонізаційний та сцинтиляційний методи (за інтенсивністю світлових спалахів, що виникають внаслідок люмінесценції в деяких речовинах під час проходження через них іонізуючих випромінювань).

При роботі з джерелами іонізуючих випромінювань здійснюють контроль і оцінку параметрів радіаційного фактора відповідно до НРБУ-97. При дотриманні контрольних рівнів умови праці на даному робочому місці оцінюються як допустимі. У разі їх перевищення оцінка шкідливості та небезпечності за радіаційним фактором здійснюється органами Держсанепіднагляду.

Засоби та заходи захисту від іонізуючих випромінювань поділяють на організаційні, технічні, санітарно-гігієнічні та лікувально-профілактичні.

Як правило, ефективний захист від іонізуючого випромінювання досягається при одночасному комплексному використанні зазначених заходів та засобів. При їх виборі враховуються особливості джерел випромінювання. Так, основними заходами, направленими на захист від альфа- та бета-випромінювань, є заходи, що націлені на недопущення накопичення альфа- і бета-активних ізотопів в організмі людини та забруднення шкіри: використання спеціального одягу та взуття, протипилових респіраторів, обезпилення повітря, вологе прибирання помешкань, недопущення вживання радіоактивно забруднених харчових продуктів, води та інші. При роботі з джерелами гама- та рентгенівського випромінювання захист персоналу досягається шляхом зниження активності джерел випромінювання, обмеження часу роботи з ними,

збільшення відстані до джерел, екранування джерела іонізуючого випромінювання або зони знаходження людини.

## ВИСНОВКИ

У ході кваліфікаційної роботи ми розглянули процеси управління в закладах харчувань.

З приходом глобалізації ринок став більш конкурентним, тому що відкрив можливості для нових конкурентів. Це не обов'язково означає ризик для виживання місцевого бізнесу, але виклик, який вони повинні розглянути. Цей виклик пов'язаний з необхідністю створення більшої лояльності споживачів до продуктів і послуг, більшої відповідності продукту потребам споживача та більшої турботи про соціальний вплив компанії. Більше того, цей глобальний сценарій представляє певні можливості для компаній діяти на нових ринках. Зрозуміло, що ця дія буде залежати в основному від якості власних продуктів і пропонуванних послуг.

Також були розглянуті методи управління проектами. Та описано метод аналізу ієрархій, який ми застосовували для вибору найкращого методу для управління процесами в ресторанному бізнесі. В ході наших обчислень ми визначили що за нашими критеріями найкращим методом є класичний проектний менеджмент.

Розробили веб-застосунок для оптимізації кавярні. Даний застосунок базується на впровадженні класичного проектного менеджменту для оптимізації роботи обраного закладу.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Пінто, Дж. (2007). Управління проектами: досягнення конкурентних переваг, Pearson Education, Великобританія
2. Аль-Харбі, К. М. (2001). Застосування АНР в управлінні проектами, Міжнародний журнал управління проектами, Vol. 19, 19-27
3. Аль Халіл, М.І. (2002). Вибір відповідного методу реалізації проекту за допомогою АНР, Міжнародний журнал управління проектами, Vol. 20, 469-474
4. Huang, CC; Чу, П.Й.; Чанг, УН (2008). Нечітка програма АНР при виборі проектів НДДКР, спонсорованих урядом, Омега, Vol. 36, 1038-1052
5. Мелоне Н.П.; Wharton, TJ (1984). Стратегії вибору проектів МІС, Журнал системного менеджменту, лютий 1984 р., 26-33
6. Шовал П.; Лугасі, Ю. (1988). Вибір комп'ютерних систем: графічний підхід "витрати-вигоди", Інформація та управління, Vol. 15, 163-172
7. Наус, А. (2002). Управління проектами, GV založba, Любляна
8. Souder, WE; Шерман, JD (1994). Управління розвитком нових технологій, Макгроу-Хілл, Нью-Йорк
9. Мередіт, молодший; Мантел, SJ (2003). Управління проектами, Вайлі, Нью-Йорк
10. Souder, WE (1988). Вибір проектів, які максимізують прибуток, Cleland, DI; Кінг, WR (редактори), Посібник з управління проектами, Ван Ностранд-Рейнхольд, 140-164
11. Раз, Т. (1997). Ітеративна методологія відбору для вибору альтернатив проекту, Журнал управління проектами, Vol. 28, № 4
12. Saaty, TL (1980). Процес аналітичної ієрархії, Макгроу-Хілл, Нью-Йорк

13. Скібнєвський М.Ю.; Чао Л. (1992). Оцінка передової технології будівництва методом АНР, Журнал інженерії будівництва та управління, Vol. 118, No 3, 577-593
14. Кремляк З.; Бухмейстер, Б. (2006). Невизначеність і розвиток можливостей, DAAAM International Publishing, Відень
15. Дольсак, Б.; Новак, М.; Єзерник, А. (2003). Інтелектуальна оптимізація дизайну на основі результатів аналізу кінцевих елементів, Міжнародний журнал передових технологій виробництва, Vol. 21, No 6, 391-396
16. Міан, С.А.; Дай, СХ (1999). Прийняття рішень протягом життєвого циклу проекту: підхід аналітичної ієрархії, Журнал управління проектами, березень 1999 р., 40-52
17. Варгас, Л.Г. (1990). Огляд процесу аналітичної ієрархії та його застосування, Європейський журнал операційних досліджень, Vol. 48, 2-8
18. Фрай, FX; Харкер, РТ (1999). Вимірювання сукупної продуктивності процесу за допомогою АНР, Європейський журнал операційних досліджень, Vol. 116, 436-442
19. Закон України «Про охорону праці» від 14 жовтня 1992 р. № 2694.
20. Довідник нормативних документів у сфері охорони праці, пожежної безпеки, гігієни праці та соціального страхування від нещасних випадків. Фонд соціального страхування від нещасних випадків на виробництві та професійних захворювань України. – К.: 2009, 244 с
21. Методи управління проектами URL:[https://vuzlit.ru/2283992/metodi\\_upravlinnya\\_proektami](https://vuzlit.ru/2283992/metodi_upravlinnya_proektami) (дата звернення: 10.12.2019).
22. Дипроуз Д. Управление проектами. Москва : «Эксмо», 2016. 240 с.

23. Капінос Г. І., Бабій І.В. Операційний менеджмент : навчальний посібник. Київ : «Центр учбової літератури», 2013. 352 с.
24. Сучасні методи управління проектами.URL: <http://diplomba.ru/work/33052> (дата звернення: 10.12.2019).
25. Сазерленд Джефф. Scrum. Революционный метод управления проектами. Москва : Манн, Иванов и Фербер, 2015. 288 с.
26. Корзаченко О.В. Оптимізація бізнес-процесів українських підприємств: проблеми та перспективи. Науковий вісник Херсонського державного університету. Економічні науки. 2013. № 3. С. 64–69.
27. Kanban. Just-In-Time at Toyota. Management Begins at the Workplace Hardcover. December, 1986. 212 p.
28. Projects IN Controlled Environments (PRINCE2) the Office of Government Commerce (OGC), United Kingdom. URL: <https://discovery.nationalarchives.gov.uk/details/r/C11669509> (дата звернення: 09.12.2019).
29. Червінська С.Л. Особливості визначення фінансових результатів діяльності аграрних підприємств. Облік і фінанси. 2009. № 6. С.107–110.
30. Управление проектами: риски. URL: <https://mipt.ru/diht/upload>
31. Тарасюк Г.М. Управління проектами : навчальний посібник для студентів.
32. Вишневська О.О. Туристичне країнознавство: підручник / О.О. Вишневська, А.Ю. Парфіненко, В.І. Сідоров. Х.: ХНУ імені В.Н. Каразіна, 2011. 594 с.
33. Гладуняк Ю.В. Державне регулювання розвитку готельного господарства: проблеми та перспективи / Ю.В. Гладуняк // Наук. вісн. Нац. Ун-ту держ. податкової служби України (економіка, право). 2009. №3. С.29-36.

34. Козлова Т. «Лицо» сетевых отелей / Т. Козлова // Новости турбизнеса. 2005. №22. С.23.
35. Мунін Г.Б. Управління сучасним готельним комплексом: навч. посіб. / Г.Б. Мунін, А.О. Змішов, Г.О. Зінов'єв, Є.В. Самарцев, О.О. Гаца; за ред. С.І. Дорогунцова. К.: Ліра, 2005. 156 с.
36. Бурков В. Н., Новиков Д. А. Как управлять проектами. – М-ИНТЕГГЕО, 1997.
37. Бушуев С.Д., Гурин Э.А. Инвестиционные инструменты проектного менеджмента. К.: Укр. ИНТЭИ, 1998. – 184с.
38. Бушуев С.Д., Морозов В.В. Управление закупками в проектах. К.: Укр. ИНТЭИ, 1999. Т.1,2– 188с., 195с.
39. Бушуев С.Д. Практика проектного менеджменту “крюк за кроком” ч 1,2. Методичні вказівки до практичних занять. КНУБА 1999 57с.,41с.
40. Виленський П.Л. Оцінка ефективності інвестиційних проектів : теорія і практика – М.: Дело , 2001 – 832 с.
41. Грей К.Ф., Ларсон Э.У. Управління проектами. – М.: Дело и Сервис, 2003. – 528 с.
42. Ефимова О.П., Ефимова Н.А. Экономика гостиниц и ресторанов. – М.: Новое знание, 2004. – 391 с.
43. Кабушкін Н.І., Бондаренко Г.А. Менеджмент готелів та ресторанів: Посібник для студ. спец. «Економіка та управління соціально-культурною сферою» навч.закладів. – 4 видан., стер. – Мінськ: ООО «Новое знание», 2003. – 368 с.
44. Кабушкін Н.І., Бондаренко Г.А. Менеджмент готелів та ресторанів: Посібник. – М.: изд-во Новое знание, 2002. – 368 с.



45. Клайэм Р., Лудін І. Ноєв проект: Секрети практичного проектного менеджменту. – СПб: Весь, 2002. – 320 с.
46. Кристофер Энертон-Томас. Ресторанный бізнес. – М. “Росконсульт” 1999г.
47. Мазур І.І., Шапіро В.Д. і інш. Управління проектами: Довідник для професіоналів. – М.: Высшая школа, 2001. – 875 с.
48. Маудер У. Вибір проекту. Планування робіт по проектам та керівництво проектами. –М.: Мир, 1991.
49. Менеджмент готельного і ресторанного обслуговування - Москва, 1997 р.
50. Світ управління проектами: основи, методи, організація, використання/За ред. Х. Решке, Х. Шелле, – М.: Аланс, 1994.
51. Нечаюк Л.І., Телеш Н.О. Готельно–ресторанний бізнес: менеджмент: Навч. посіб. для студ. вищих навч. закл. / Центр навчальної літератури; Київський національний ун-т культури і мистецтв. – К.: Центр Навчальної Літератури, 2003. – 346 с.
52. Пинто Дж.К. Управление проектами. – СПб: Питер, 2004. – 464 с.
53. Проектний аналіз . Навч. посібн. / За ред. Москвіна С.О. – К. 1999 – 368с.
54. Турчин С.А., Программный инструментарий менеджера проекта, Киев, Компьютерное обозрение №34/2000 - с. 15-20
55. Управление программами и проектами / М.Л.Разу и др. – М: ИНФРА, 1999.
56. Управление проектами / Под ред. В.Д.Шапира - М., 1996 – 610 с.
57. Управление проектами / Под ред. Шапира. – СПб: Два-Три, 1996.

58. Управление проектами: зарубежный опыт / А. И. Кочетков и др. – СПб.: Два-Три, 1993.
59. Усов В.В. «Организация обслуживания в ресторане» - Москва. «Высшая школа» 1990г.
60. Устинов А.В. Экономика материально-технического снабжения. – М.: Высшая школа – 1994 – 315 с

# ДОДАТКИ

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ІВАНА ПУЛЮЯ**

**МАТЕРІАЛИ**

**ІХ НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ**

**«ІНФОРМАЦІЙНІ МОДЕЛІ,  
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



**8–9 грудня 2021 року**

**ТЕРНОПІЛЬ  
2021**

<b>Р.І. Боднар, І.М. Кормило, О.Ю. Задолнний, Т.О. Масвський</b> СИСТЕМИ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ОПРАЦЮВАННЯ ДАНИХ В УМОВАХ ПАНДЕМІЇ	
<b>R.I. Bodnar, I.M. Kormylo, O.Yu. Zadolynnyi, T.O. Maievskyi</b> ARTIFICIAL INTELLIGENCE SYSTEMS FOR DATA PROCESSING IN A PANDEMIC CONDITON	29
<b>Р.І. Боднар, І.М. Кормило, О.Ю. Задолнний, Т.О. Масвський</b> ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ ДЛЯ ОПРАЦЮВАННЯ ДАНИХ В УМОВАХ ПАНДЕМІЇ	
<b>R.I. Bodnar, I.M. Kormylo, O.Yu. Zadolynnyi, T.O. Maievskyi</b> CONVOLUTIONAL NEURAL NETWORKS FOR DATA PROCESSING IN A PANDEMIC CONDITON	31
<b>А.І. Войтович</b> ДОСЛІДЖЕННЯ АКТУАЛЬНОСТІ УПРАВЛІННЯ ПРОЕКТАМИ У СФЕРІ ОБСЛУГОВУВАННЯ	33
<b>Р.І. Волощак</b> РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ЗБОРУ ТА АНАЛІЗУ ПОКАЗНИКІВ ЛІЧИЛЬНИКА ЕЛЕКТРОЕНЕРГІЇ З ВИКОРИСТАННЯМ ARDUINO	
<b>R.I. Voloshchak</b> DEVELOPMENT OF INFORMATION SYSTEM FOR COLLECTION AND ANALYSIS OF ELECTRICITY METER INDICATORS USING ARDUINO	34
<b>О.В. Волянник, Ннамене Крістофер Чізоба, С.А. Лупенко</b> ПРОТОТИП ІНФОРМАЦІЙНОЇ ОНТООРІЄНТОВАНОЇ ДОВІДКОВОЇ СИСТЕМИ ПРЕДМЕТНОЇ ОБЛАСТІ «МОДЕЛЮВАННЯ ТА ОПРАЦЮВАННЯ ЦИКЛІЧНИХ СИГНАЛІВ»	
<b>O.V. Volyanyk, Nnamene Christopher Chizoba, S.A.</b> PROTOTYPE OF ONTO-ORIENTED INFORMATION HELP SYSTEM IN SUBJECT AREA «MODELING AND PROCESSING CYCLIC SIGNALS»	35
<b>А.О. Воронка</b> МОДЕЛЬ ПАМ'ЯТІ ТЕХНОЛОГІЇ CUDA	
<b>A.O. Voronka</b> CUDA TECHNOLOGY MEMORY MODEL	36
<b>А.В. Гайдар, В.А. Готович</b> РОЗРОБКА ПЛАТФОРМИ ДЛЯ ПЕРЕВІРКИ ЗНАНЬ ШЛЯХОМ ТЕСТУВАННЯ	
<b>A.V. Haidar, V.A. Gotovych</b> DEVELOPMENT OF PLATFORMS FOR VERIFICATION OF KNOWLEDGE THROUGH TESTING	37
<b>Ю. Горбуляк</b> ОГЛЯД МЕТОДІВ МАЙНІНГУ WEB-КОНТЕНТУ	
<b>Yu. Horbuliak</b> SURVEY OF THE METHODS OF WEB-CONTENT MINING	38
<b>С. Гошко, Г.В. Козбур</b> ВИКОРИСТАННЯ ВЕЛИКИХ ДАНИХ В РОЗУМНОМУ МІСТІ	
<b>E. HotskoH. Kozbur</b> USING BIG DATA IN A SMART CITY	39

<b>О. Ревнюк, Н. Загородна</b> МОДЕЛІ ТА МЕТОДИ ОЦІНЮВАННЯ ЯКОСТІ ВЕБ-ДОДАТКІВ <b>O. Revnuk, N. Zagorodna</b> MODELS AND METHODS FOR EVALUATION OF WEB-APPLICATIONS QUALITY	73
<b>А.В. Степанов, А.А. Микитишин</b> ДОСЛІДЖЕННЯ ТА УДОСКОНАЛЕННЯ СТАНДАРТНИХ МЕТОДІВ ЗАХИСТУ ВЕБ-ДОДАТКІВ <b>A.V. Stepanov, A.A. Mykytyshyn</b> STUDY AND IMPROVEMENT OF STANDARD METHODS OF WEB- APPLICATIONS PROTECTION	75
<b>М.М. Серватнюк</b> ІНТЕГРАЦІЯ МЕТОДІВ OSINT В СИСТЕМУ УПРАВЛІННЯ ІНФОРМАЦІЙНИМ РИЗИКАМИ <b>M.M. Servatniuk</b> INTEGRATION OF OSINT METHODS INTO THE INFORMATION RISK MANAGEMENT SYSTEM	76
<b>Д. Сіренко, Р. Грималовський, І. Дедів</b> СПОСІБ ПІДСИЛЕННЯ ВІЗУАЛЬНОГО СПРИЙНЯТТЯ ЗОБРАЖЕНЬ НА ОСНОВІ СТОХАСТИЧНОГО РЕЗОНАНСУ <b>D. Sirenko, R. Hrymalovsky, I. Dediv</b> THE METHOD OF ENHANCING VISUAL PERCEPTION OF IMAGES ON THE BASIS OF STOCHASTIC RESONANCE	77
<b>Д.О. Стьопа, О.М. Ярема</b> МЕТОДИ ЗАХИСТУ ІНФОРМАЦІЙНО-ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ ТА МЕРЕЖ ВІД НЕСАНКЦІОНОВАНОГО ДОСТУПУ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ VPN	78
<b>В.В. Сумко</b> ОСОБЛИВОСТІ ЕКСПЕРТНИХ СИСТЕМ У ПОРІВНЯННІ ЗІ ЗВИЧАЙНИМИ КОМП'ЮТЕРНИМИ ПРОГРАМАМИ <b>V.V. Sumko</b> REGULARITIES OF EXPERT SYSTEMS IN COMPARISON WITH COMMON COMPUTER PROGRAMS	79
<b>В.В. Гафінєць, І.В. Струтинська</b> БЛОКЧЕЙН ТА РОЗУМНЕ МІСТО <b>V. Hafinets, I. Strutynska</b> BLOCKCHAIN AND A SMART CITY	80
<b>І.В. Станько, А.І. Войтович</b> ОГЛЯД МЕТОДІВ ОПТИМІЗАЦІЇ ПРОЦЕСІВ <b>I. Stanko, A. Vojtovych</b> OVERVIEW OF PROCESS OPTIMIZATION METHODS	81
<b>Ю.ІІ. Снібайло</b> ОГЛЯД СИСТЕМИ ПІД НАЗВОЮ K COMPUTER <b>Y. Stsibailo</b> OVERVIEW OF THE K COMPUTER SYSTEM	83



УДК 004.4'2

**А.І. Войтович**

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

## ДОСЛІДЖЕННЯ АКТУАЛЬНОСТІ УПРАВЛІННЯ ПРОЕКТАМИ У СФЕРІ ОБСЛУГОВУВАННЯ

Менеджери ресторану відповідають за щоденну роботу ресторану, який готує та подає страви та напої клієнтам. Сильні навички планування, особливо координація роботи з різними відділами (кухня, їдальня, банкетні операції, менеджери громадського харчування, постачальники, що надають товари), гарантують, що клієнти будуть задоволені своїм обідом. Невід'ємним навичком є здатність менеджерів залучати й утримувати працівників, а також контролювати ефективність і навчання працівників забезпечуючи якість із стримуванням витрат. Планування в багатьох аспектах, не тільки в роботі персоналу, але й у термінах доставки продуктів харчування, має вирішальне значення для задоволення очікувань клієнтів [1].

Управління ризиками є важливим для забезпечення безпеки та якості харчових продуктів. Менеджери контролюють замовлення на кухні, щоб визначити, де можуть виникнути затримки, і вони співпрацюють з шеф-кухарем, щоб уникнути цих затримок. Для того, щоб ресторан залишався відкритим, важливо дотримуватися законодавчих норм, тому керівники ресторану керують прибиранням обідніх зон та миттям столового посуду, кухонного начиння та обладнання. Вони забезпечують дотримання стандартів безпеки та законності, особливо при подачі алкоголю. Чутливість і сильні комунікативні навички необхідні, коли клієнти скаржаться або працівники відчувають тиск, оскільки приходить більше клієнтів, ніж передбачалося.

Фінансові знання необхідні для надійного функціонування ресторану, особливо для відстеження спеціальних проєктів, подій та витрат на різноманітні меню [2]. Безперерйне проведення заходів з громадського харчування може бути результатом використання планів проєкту та філософії управління проєктами. Менеджер ресторану або шеф-кухар аналізують рецепти, щоб визначити витрати на їжу, робочу силу та накладні витрати; визначає розмір порції та поживність кожної порції; і призначає ціни на різні пункти меню, щоб товари можна було замовити та отримати вчасно.

Планування є запорукою успішної реалізації. Менеджери або шеф-кухари повинні оцінювати потреби в їжі, розміщувати замовлення у дистриб'юторів і планувати доставку свіжих продуктів і запасів. Вони також планують регулярні послуги (обслуговування обладнання, боротьба з шкідниками, вивезення відходів) та доставку, включаючи послуги прання скатертин або інтенсивне прибирання залу чи кухонного обладнання, що відбуватися коли заклад зачинений. Успішний ресторан покладається на багато чинників, на яких наголошує професія управління проєктами.

Для покращення та автоматизації роботи таких закладів, пропонується розробка АСУ, яка буде володіти функціями планування ресурсів, управління ризиками, персоналом.

### Література.

1. Кузьмін О.В. Інжиніринг в ресторанному бізнесі/Навчальний посібник. – О.В. Кузьмін, О.В. Чемакіна, Л.М. Акімова. – ОЛДІ-ПЛЮС. – 2019. – 485 с.
2. Андренко І.Б. Менеджмент готельно-ресторанного господарства: підручник / І.Б. Андренко, О.М. Кравець, Л.М. Писаревський ; Харк. нац. ун-т міськ. госп-ва ім. О.М. Бекетова. – Х.: ХНУМГ, 2014. – 431 с.

УДК 004.415.5

**I.В. Станько, А.І. Войтович**

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

## ОГЛЯД МЕТОДІВ ОПТИМІЗАЦІЇ ПРОЦЕСІВ

УДК 004.415.5

**I. Stanko, A. Vojtovych**

## OVERVIEW OF PROCESS OPTIMIZATION METHODS

Методи оптимізації процесів.

Загалом, оптимізація процесу не є синонімом оптимізації, а вдосконалення. Таким чином, методи оптимізації процесів насправді є заходами, які спрямовані на досягнення покращень. Якщо раніше оптимізація процесів забезпечувалась управлінням якістю, то тепер це завдання все більше покладається на промислове інженерство. Є кілька способів покращення процесів. Важливо запровадити методи оптимізації процесу та правильно проаналізувати відповідні умови. Методи оптимізації процесу можуть бути досягнуті лише за допомогою структурованого підходу. Існує багато способів оптимізації процесів [1].

Підвищення продуктивності можна досягти кількома способами. Використовуються технічні та організаційні засоби, оскільки вони дозволяють досягти значно більшого, ніж один із цих підходів у поєднанні. Як допоміжні засоби можна використовувати технічні засоби або навіть автоматику. Технічні засоби повинні відповідати заданій структурі. Наприклад, купувати повністю автоматичного зварювального робота не варто, якщо потрібно зварити лише кілька деталей. Але оптимізація процесів за допомогою автоматизації є важливим фактором, який не слід недооцінювати. Метод 5S може бути оптимізацією процесів або підходом до покращення процесів на робочому місці і, отже, підвищення продуктивності. Метод 5S є одним із можливих методів покращення процесів на робочому місці і, таким чином, підвищення продуктивності. П'ять «S» спочатку походять від японських термінів Seiri, Seiton, Seiso, Seiketsu та Shitsuke, які представляють п'ять кроків методу 5S. У перекладі вони означають: відбирати, систематизувати, очищати, стандартизувати і самодисциплінувати.

Оптимізація процесу: процедура вдосконалення виробничих ділянок.

Планування розташування є важливою частиною планування заводу. Макет зазвичай відображається за допомогою системи САД. Зменшення площі та розміщення послуг один щодо одного планується спеціальними методами. Таким чином виходить ідеальна оптимізація процесу. Трикутний метод за Шмігаллою може дати дуже хороші результати. Цей метод дозволяє оптимально координувати послуги. Важливим є потік матеріалів між різними відділами [1].

Оптимізація процесу: процедура покращення використання потужностей

Використання машин можна покращити кількома способами. Дуже ефективним рішенням є скорочення часу приготування. Це один із методів бережливого виробництва методів оптимізації процесів.

Оптимізація процесу: процедура вдосконалення процесу

Щоб удосконалити процеси, необхідно їх попередньо проаналізувати. Існує ряд методів оптимізації процесу, які можна застосувати в конкретному випадку. Існує велика різноманітність методів оптимізації процесів і варіантів використання.

Оптимізація процесу: процедура скорочення шляху

Скорочення часу в дорозі може вплинути на робочу систему або на всю компанію. У робочій системі зменшення переміщень можна досягти за допомогою діаграми спагетті. Ця оптимізація процесу/процедур не може бути застосована до всієї компанії. Планування розвитку, наприклад, за методом трикутників за Шмігаллою, однак також покращує шляхи. Якщо на підприємстві доводиться перевозити менше матеріалу, відстань, яку необхідно подолати, також зменшується [2].



#### Оптимізація процесу: процедура підвищення якості

Хороша якість є одним із найважливіших методів оптимізації процесів: якість завжди відігравала особливу роль, а в промислово розвиненій країні, зокрема, фактор якості є вирішальним фактором. Якість визначається за допомогою вимірювального та випробувального обладнання, але якість створюється набагато раніше, і для цього необхідно створити відповідні рамки. Метод Six Sigma може бути основою для забезпечення якості. Оптимізація процесу: процедура зниження витрат

Перераховані вище методи сприяють зниженню витрат, тому не існує конкретної оптимізації процесу/підходу, який сам по собі знижує витрати. У більшості випадків існує багато методів оптимізації процесів, які використовуються для обмеження витрат, які несе бізнес. Немає конкретної оптимізації/підходу процесів, який сам по собі здатний покращити все в бізнесі. Коли мова йде про ізольовану зону, завжди використовуються кілька методів оптимізації процесів, у комбінації або окремо.

#### Висновок: методи оптимізації процесу

Методи оптимізації процесу: для досягнення мети важливо правильно проаналізувати початкову ситуацію та вжити відповідних дій за допомогою правильних методів оптимізації процесу. Поєднання жодна оптимізація процесу/процедури не досягає мети, і, використовуючи методи, ви повинні знати, як вони працюють і чого можна навчитися з них. Тільки точне знання методів та їх потенціалу дозволяє їх правильно використовувати. На жаль, часто буває, що методи оптимізації процесів недостатньо зрозумілі. Але фатальна річ настає пізніше, тому що тоді функціональність методів оптимізації процесів також ігнорується. Багато методів Оптимізація процесів розвиває свій потенціал лише в поєднанні з іншими методами оптимізації процесів. Використання єдиної оптимізації процесу/процедури без розуміння цих взаємозв'язків неминуче призводить до провалу зусиль. Успіх приходить від тих, хто правильно застосовує експертні знання та правильно використовує методи оптимізації процесів. З MotionMiners GmbH у вас є не тільки потужний інструмент, який автоматично записує та аналізує процеси, а й досвідчені інженери-технологи, які допоможуть вам вибрати правильні методи оптимізації процесів. єдиний процес оптимізації/процедури без розуміння цих взаємозв'язків неминуче призводить до провалу зусиль. Успіх приходить від тих, хто правильно застосовує експертні знання та правильно використовує методи оптимізації процесів. З MotionMiners GmbH у вас є не тільки потужний інструмент, який автоматично записує та аналізує процеси, але й досвідчені інженери-технологи, які допоможуть вам вибрати правильні методи оптимізації процесів.

Ми розглянули різні методи оптимізації, так як наша робота в першу чергу побудована для оптимізації процесів.

#### Література.

1. Кузьмін О.С. Сучасний менеджмент. – Львів, 1995.
3. Мескон М.Х., Альберт М., Хедоури Ф. Основы менеджмента / Пер. с англ. – М., 1998.

## Код програми

## index.html

## &lt;!DOCTYPE html&gt;

```

<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/fflogo-red.ico" />
    <meta
      name="viewport"
      content="minimum-scale=1, initial-scale=1, width=device-width"
    />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <!--
      manifest.json provides metadata used when your web app is installed on a
      user's mobile device or desktop. See
      https://developers.google.com/web/fundamentals/web-app-manifest/
    -->
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <!--
      Notice the use of %PUBLIC_URL% in the tags above.
      It will be replaced with the URL of the `public` folder during the build.
      Only files inside the `public` folder can be referenced from the HTML.

      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
      work correctly both with client-side routing and a non-root public URL.
      Learn how to configure a non-root public URL by running `npm run build`.
    -->
    <link
      rel="stylesheet"
      href="https://fonts.googleapis.com/icon?family=Material+Icons"
    />
    <title>FastFruit</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
    <!--
      This HTML file is a template.
      If you open it directly in the browser, you will see an empty page.

      You can add webfonts, meta tags, or analytics to this file.
      The build step will place the bundled scripts into the <body> tag.

      To begin the development, run `npm start` or `yarn start`.
      To create a production bundle, use `npm run build` or `yarn build`.
    -->

```

```
</body>
</html>
```

## serviceWorker.js

```
// This optional code is used to register a service worker.
// register() is not called by default.

// This lets the app load faster on subsequent visits in production, and gives
// it offline capabilities. However, it also means that developers (and users)
// will only see deployed updates on subsequent visits to a page, after all the
// existing tabs open on the page have been closed, since previously cached
// resources are updated in the background.

// To learn more about the benefits of this model and instructions on how to
// opt-in, read https://bit.ly/CRA-PWA

const isLocalhost = Boolean(
  window.location.hostname === 'localhost' ||
  // [::1] is the IPv6 localhost address.
  window.location.hostname === '[:1]' ||
  // 127.0.0.0/8 are considered localhost for IPv4.
  window.location.hostname.match(
    /^127(?:\.(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?) ){3}$/
  )
);

export function register(config) {
  if (process.env.NODE_ENV === 'production' && 'serviceWorker' in navigator) {
    // The URL constructor is available in all browsers that support SW.
    const publicUrl = new URL(process.env.PUBLIC_URL, window.location.href);
    if (publicUrl.origin !== window.location.origin) {
      // Our service worker won't work if PUBLIC_URL is on a different origin
      // from what our page is served on. This might happen if a CDN is used to
      // serve assets; see https://github.com/facebook/create-react-app/issues/2374
      return;
    }

    window.addEventListener('load', () => {
      const swUrl = `${process.env.PUBLIC_URL}/service-worker.js`;

      if (isLocalhost) {
        // This is running on localhost. Let's check if a service worker still exists or
        // not.
        checkValidServiceWorker(swUrl, config);

        // Add some additional logging to localhost, pointing developers to the
        // service worker/PWA documentation.
        navigator.serviceWorker.ready.then(() => {
```

```

        console.log(
            'This web app is being served cache-first by a service ' +
            'worker. To learn more, visit https://bit.ly/CRA-PWA'
        );
    });
} else {
    // Is not localhost. Just register service worker
    registerValidSW(swUrl, config);
}
});
}
}

function registerValidSW(swUrl, config) {
    navigator.serviceWorker
        .register(swUrl)
        .then(registration => {
            registration.onupdatefound = () => {
                const installingWorker = registration.installing;
                if (installingWorker == null) {
                    return;
                }
                installingWorker.onstatechange = () => {
                    if (installingWorker.state === 'installed') {
                        if (navigator.serviceWorker.controller) {
                            // At this point, the updated precached content has been fetched,
                            // but the previous service worker will still serve the older
                            // content until all client tabs are closed.
                            console.log(
                                'New content is available and will be used when all ' +
                                'tabs for this page are closed. See https://bit.ly/CRA-PWA.'
                            );

                            // Execute callback
                            if (config && config.onUpdate) {
                                config.onUpdate(registration);
                            }
                        } else {
                            // At this point, everything has been precached.
                            // It's the perfect time to display a
                            // "Content is cached for offline use." message.
                            console.log('Content is cached for offline use.');
                        }

                        // Execute callback
                        if (config && config.onSuccess) {
                            config.onSuccess(registration);
                        }
                    }
                }
            }
        })
        .catch(error => {
            console.error('Error during service worker registration:', error);
        });
}
}

```

```

function checkValidServiceWorker(swUrl, config) {
  // Check if the service worker can be found. If it can't reload the page.
  fetch(swUrl, {
    headers: { 'Service-Worker': 'script' },
  })
  .then(response => {
    // Ensure service worker exists, and that we really are getting a JS file.
    const contentType = response.headers.get('content-type');
    if (
      response.status === 404 ||
      (contentType !== null && contentType.indexOf('javascript') === -1)
    ) {
      // No service worker found. Probably a different app. Reload the page.
      navigator.serviceWorker.ready.then(registration => {
        registration.unregister().then(() => {
          window.location.reload();
        });
      });
    } else {
      // Service worker found. Proceed as normal.
      registerValidSW(swUrl, config);
    }
  })
  .catch(() => {
    console.log(
      'No internet connection found. App is running in offline mode.'
    );
  });
}

export function unregister() {
  if ('serviceWorker' in navigator) {
    navigator.serviceWorker.ready
      .then(registration => {
        registration.unregister();
      })
      .catch(error => {
        console.error(error.message);
      });
  }
}

```

## App.js

```

import React, { useState, useEffect, useRef } from "react";
import { BrowserRouter, Route, Switch } from "react-router-dom";
import { Navigation } from "./components/Navigation";
import Home from "./components/Home";
import Order from "./components/Order";
import { Register } from "./components/Register";
import { Login } from "./components/Login";
import { AdminDashboard } from "./components/AdminDashboard";
import EmployeeDashboard from "./components/EmployeeDashboard";
import { FoodMenu } from "./components/FoodMenu";
import { ToastContainer, toast } from "react-toastify";
import { OrderListContext } from "./components/context/Context";
import "react-toastify/dist/ReactToastify.css";
import "./css/App.css";
import socketIOClient from "socket.io-client";

```

```

import axios from "axios";
import fb from "./img/fb.png";
import tw from "./img/tw.png";
import ig from "./img/ig.png";

let socket;

function App() {
  const [orderList, setOrderList] = useState(true);
  const [notifyNewOrder, setNotifyNewOrder] = useState(true);
  const [notifyOrderReady, setNotifyOrderReady] = useState("client_id");
  const isFirstRunNewOrder = useRef(true);
  const isFirstRunOrderReady = useRef(true);

  useEffect(() => {
    console.log("CAMBIA");
    getUserCredentials().then((user) => {
      getUserOrdersIds(user.id).then((ids) => {
        // Start a web socket globally
        socket = socketIOClient("/");

        // Inform an employee when a new order has been placed by a client
        socket.on("inform-employees", (data) => {
          if (user.type === "admin" || user.type === "employee")
            toast.success(data.message, {
              position: toast.POSITION.BOTTOM_RIGHT,
              className: "ff-notice-toast",
              autoClose: false,
            });
        });

        // Inform a client when his order has been prepared by an employee
        socket.on("inform-client", (data) => {
          if (
            (user.type === "admin" ||
              user.type === "employee" ||
              user.type === "client") &&
            ids.includes(parseInt(data.id))
          )
            toast.success(data.message, {
              position: toast.POSITION.BOTTOM_RIGHT,
              className: "ff-ready-toast",
              autoClose: false,
            });
        });
      });
    });
  });

  }, [orderList]);

  useEffect(() => {
    if (isFirstRunNewOrder.current) {
      isFirstRunNewOrder.current = false;
      return;
    }
  });

```

```

    }

    socket.emit("new-order-placed", { message: "New order has been placed" });
  }, [notifyNewOrder]);

useEffect(() => {
  if (isFirstRunOrderReady.current) {
    isFirstRunOrderReady.current = false;
    return;
  }

  socket.emit("order-ready", {
    message: "An order has been marked as ready",
    id: notifyOrderReady,
  });
}, [notifyOrderReady]);

return (
  <OrderListContext.Provider value={[orderList, setOrderList]}>
    <BrowserRouter>
      <Navigation />
      <div className="container">
        <Switch>
          <Route path="/" component={Home} exact />
          <Route path="/FoodMenu" component={FoodMenu} />
          <Route
            path="/client/order"
            render={(props) => (
              <Order
                notifyNewOrder={notifyNewOrder}
                setNotifyNewOrder={setNotifyNewOrder}
              />
            )}
          />
          <Route path="/register" component={Register} />
          <Route path="/login" component={Login} />
          <Route path="/admin/tools" component={AdminDashboard} />
          <Route
            path="/employee/tools"
            render={(props) => (
              <EmployeeDashboard
                notifyOrderReady={notifyOrderReady}
                setNotifyOrderReady={setNotifyOrderReady}
              />
            )}
          />
        </Switch>
      </div>
      <footer className="bg-ff-footer text-white address">
        <div className="container">
          <div className="row">
            <div className="col-lg-3 col-md-6 col-sm-6 col-xs-12 mr-5">
              <ul>
                <span>Authors</span>
                <li>Kevin Ruvalcaba P.</li>
                <li>Alejandro Moreno L.</li>
                <li>Germán Reyes G.</li>
              </ul>
            </div>
          </div>
        </div>
      </footer>
    </OrderListContext.Provider>
  )
)

```

```

    </ul>
  </div>
  <div className="col-lg-3 col-md-6 col-sm-6 col-xs-12 mr-5">
    <ul className="social">
      <span>Project's repository</span>
      <li>
        <a href="https://github.com/germanreyga/Restaurant-app">
          
        </a>
      </li>
    </ul>
  </div>
  <div className="col-lg-3 col-md-6 col-sm-6 col-xs-12">
    <ul className="social">
      <span>Social media</span>
      <li>
        <img alt="Facebook" src={fb} className="social-icons" />
      </li>
      <li>
        <img alt="twitter" src={tw} className="social-icons" />
      </li>
      <li>
        <img alt="instagram" src={ig} className="social-icons" />
      </li>
    </ul>
  </div>
</div>
</div>
<ToastContainer />
</footer>
</BrowserRouter>
</OrderListContext.Provider>
);

async function getUserCredentials() {
  let type = undefined;
  let id = undefined;
  await axios
    .get("/user/credentials")
    .then((res) => {
      type = res.data.type;
      id = res.data.id;
    })
    .catch((err) => {
      console.log(err);
    });

  const user = {
    id: id,
    type: type,
  };
  return user;
  //setCredentials(user);
}

```



```

async function getUserOrdersIds(id) {
  let ids = [];
  await axios
    .get(`/order/all/${id}`)
    .then((res) => {
      ids = res.data.ids;
    })
    .catch((err) => {
      console.log(err);
    });

  const numericalIds = ids.map((object) => {
    return object.id_order;
  });

  return numericalIds;
  //setOrdersIds(numericalIds);
}
}

export default App;

```

## index.js

```

import React from "react";
import ReactDOM from "react-dom";
import "bootstrap/dist/css/bootstrap.min.css";
import "./css/index.css";
import App from "./App";
import * as serviceWorker from "./serviceWorker";

```

```

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById("root")
);

```

```

// If you want your app to work offline and load faster, you can change
// unregister() to register() below. Note this comes with some pitfalls.
// Learn more about service workers: https://bit.ly/CRA-PWA
serviceWorker.unregister();

```

## AdminDashboard.js

```

import React, { Component } from "react";
import { Button, Form, Alert, Table } from "react-bootstrap";
import axios from "axios";

```

```

export class AdminDashboard extends Component {
  constructor(props) {
    super(props);
    this.handleChange = this.handleChange.bind(this);
    this.handleSubmit = this.handleSubmit.bind(this);
  }

  state = {
    inputType: "employee",
    inputStore: "-",
    errors: [],
    success: undefined,
    employees: [],
    stores: [],
  };

  async componentDidMount() {
    await this.getAllStores();
    await this.getAllEmployees();
  }

  async getAllStores() {
    await axios
      .get("/stores/all")
      .then((res) => {
        this.setState({ stores: res.data.data });
      })
      .catch((err) => console.log(err));
  }

  async getAllEmployees() {
    await axios
      .get("/employees/all")
      .then((res) => {
        this.setState({ employees: res.data.data });
      })
      .catch((err) => console.log(err));
  }

  handleSubmit(event) {
    event.preventDefault();
    this.createUser(this.state);
  }

  async createUser(userinfo) {
    let user = {
      inputUsername: userinfo.inputUsername,
      inputType: userinfo.inputType,
      inputStore: userinfo.inputStore,
      inputPassword: userinfo.inputPassword,
      inputConfirmPassword: userinfo.inputConfirmPassword,
    };

    axios({

```

```

    method: "post",
    url: "/registerUser",
    data: user,
  })
  .then((res) => {
    if (res.data.status !== undefined) {
      this.setState({
        success: undefined,
        errors: res.data.message.errors,
      });
    } else {
      this.getAllEmployees();
      this.setState({ success: res.data.message, errors: [] });
    }
  })
  .catch((err) => {
    console.log(err);
    this.setState({ success: undefined, errors: err.description });
    console.log(err);
  });
}

```

```

async handleInputChange(event) {
  // This method lets us control multiple inputs in a form.
  // When an input in a form changes, this.state also updates
  // to form a single source of truth
  const target = event.target;
  const value = target.value;
  const name = target.name;

  await this.setState({
    [name]: value,
  });
}

```

```

render() {
  return (
    <div className="container">
      <br />
      <h3>
        <span role="img" aria-label="employee">
          👤
        </span>
        Add employee
      </h3>
      <hr />
      <EmployeeForm
        onSubmit={this.handleSubmit}
        state={this.state}
        handleInputChange={this.handleInputChange}
      />
      <br />
      <h3>
        <span role="img" aria-label="employees">
          👥
        </span>
        Employee list
      </h3>
    </div>
  );
}

```

```

    <hr />
    <EmployeeList employees={this.state.employees} />
  </div>
);
}
}

function EmployeeForm(props) {
  return (
    <Form className="form-signin shadow-lg p-3" onSubmit={props.onSubmit}>
      <div className="form-label-group">
        <label htmlFor="inputUsername">Username</label>
        <input
          id="inputUsername"
          name="inputUsername"
          className="form-control"
          value={props.state.username}
          onChange={props.handleInputChange}
          required
        />
      </div>
      <Form.Group>
        <label htmlFor="inputType">Type</label>
        <select
          id="inputType"
          name="inputType"
          className="form-control"
          value={props.state.type}
          onChange={props.handleInputChange}
          required
          defaultValue="-"
        >
          <option value="-" disabled>
            ---
          </option>
          <option value="employee">Employee</option>
          <option value="admin">Administrator</option>
        </select>
      </Form.Group>

      <div className="form-label-group">
        <label htmlFor="inputStore">Related store</label>
        <select
          id="inputStore"
          name="inputStore"
          className="form-control"
          defaultValue={props.state.inputStore}
          onChange={props.handleInputChange}
          required
        >
          <option value="-" disabled>
            ---
          </option>
          <StoreList stores={props.state.stores}></StoreList>
        </select>
      </div>

      <div className="form-label-group">
        <label htmlFor="inputPassword">Password</label>

```

```

    <input
      type="password"
      id="inputPassword"
      name="inputPassword"
      className="form-control"
      value={props.state.password}
      onChange={props.handleInputChange}
      required
    />
  </div>
  <div className="form-label-group">
    <label htmlFor="inputConfirmPassword">Confirm password</label>
    <input
      type="password"
      id="inputConfirmPassword"
      name="inputConfirmPassword"
      className="form-control"
      value={props.state.hashpassword}
      onChange={props.handleInputChange}
      required
    />
  </div>
  <br />
  <Success success={props.state.success} />
  <Errors errors={props.state.errors} />
  <br />
  <Button className="btn btn-lg btn-green btn-block" type="submit">
    Add this employee
  </Button>
</Form>
);
}

function Success(props) {
  if (props.success !== undefined) {
    return <Alert variant="success">User created successfully</Alert>;
  } else {
    return <div></div>;
  }
}

function Errors(props) {
  if (!props.errors || props.errors.length === 0) {
    return <div></div>;
  }
  return (
    <Alert variant="danger">
      Oops! the following errors were found:
      <ul>
        {props.errors.map((value, index) => {
          return <li key={index}>{value.msg}</li>;
        })}
      </ul>
    </Alert>
  );
}

function StoreList(props) {
  const stores = props.stores;

```

```

const listStores = stores.map((value, index) => {
  return (
    <option key={value.id_store} value={value.id_store}>
      {value.name} ({value.location})
    </option>
  );
});

return <React.Fragment>{listStores}</React.Fragment>;
}

function EmployeeList(props) {
  const employees = props.employees;

  if (employees.length > 0) {
    const listEmployees = employees.map((value, index) => {
      return (
        <tr key={value.id_user}>
          <td>{value.id_user}</td>
          <td>{value.username}</td>
          <td>{value.type}</td>
          <td>{value.id_store}</td>
        </tr>
      );
    });

    return (
      <Table striped bordered hover size="sm" className="text-center shadow-lg">
        <thead>
          <tr>
            <th>ID</th>
            <th>Username</th>
            <th>Type</th>
            <th>Related store id</th>
          </tr>
        </thead>
        <tbody>{listEmployees}</tbody>
      </Table>
    );
  } else {
    return (
      <React.Fragment>
        <div>No employees in the database</div>
        <br />
      </React.Fragment>
    );
  }
}

```

## EmployeeDashboard.js

```

import React, { useState, useEffect } from "react";
import { Card, Alert, CardDeck, Table, Button, Form } from "react-bootstrap";
import axios from "axios";

```

```

function EmployeeDashboard(props) {
  const [preparingOrdersList, setPreparingOrdersList] = useState([]);
  const [otherOrdersList, setOtherOrdersList] = useState([]);
  const setNotifyOrderReady = props.setNotifyOrderReady;

  useEffect(() => {
    const firstLoad = async () => {
      await getPreparingOrders().then((list) => {
        setPreparingOrdersList(list);
      });

      await getReadyOrDelivOrdersList().then((list) => {
        setOtherOrdersList(list);
      });
    };

    firstLoad();
  }, []);

  const pendingOrderSubmit = async (event) => {
    event.preventDefault();
    event.persist();

    const id = event.target.id.value; // Gets the input with the id_order

    await axios
      .post(`/order/ready/${id}`)
      .then((res) => {
        setNotifyOrderReady(res.data.order_id);
      })
      .catch((err) => console.log(err));

    reloadData();
  };

  const otherOrderSubmit = async (event) => {
    event.preventDefault();
    event.persist();

    const id = event.target.id.value; // Gets the input with the id_order

    await axios
      .post(`/order/delivered/${id}`)
      .then((res) => {
        console.log(res);
      })
      .catch((err) => console.log(err));

    reloadData();
  };

```

```

};

const getPreparingOrders = async () => {
  let newList = [];
  await axios
    .get("/order/preparing")
    .then((res) => {
      const ids = res.data.data;

      ids.map(async (item, index) => {
        await axios
          .get(`/order/products/${item.id_order}`)
          .then((res) => {
            newList.push({ order: res.data.data });
          })
          .catch((err) => {
            console.log(err);
          });
      });
    })
    .catch((err) => {
      console.log(err);
    });

  return newList;
};

const getReadyOrDelivOrdersList = async () => {
  let list = [];
  await axios
    .get("/order/readyOrDelivered")
    .then((res) => {
      list = res.data.data;
    })
    .catch((err) => console.log(err));
  return list;
};

const reloadData = async () => {
  await getPreparingOrders().then((list) => {
    setPreparingOrdersList(list);
  });

  await getReadyOrDelivOrdersList().then((list) => {
    setOtherOrdersList(list);
  });
};

return (
  <div className="container">
    <br />
    <h3>
      <span role="img" aria-label="pending-order">

```



```

    </span>
    Orders to prepare
  </h3>
  <hr />
  <PendingOrders
    orders={preparingOrdersList}
    onSubmit={pendingOrderSubmit}
  />
  <br />
  <h3>
    <span role="img" aria-label="ready-order">
      ↑
    </span>
    Completed orders
  </h3>
  <hr />
  <OtherOrders orders={otherOrdersList} onSubmit={otherOrderSubmit} />
</div>
);
}

```

```

function PendingOrders(props) {
  const orders = props.orders;
  if (orders.length !== undefined && orders.length > 0) {
    const orderList = orders.map((outside, index) => {
      const orderListInside = outside.order.map((inside, index) => {
        return (
          <div className="text-center border-bottom" key={index}>
            {inside.name} x {inside.quantity}
          </div>
        );
      });
    });
    return (
      <Card
        border="dark"
        key={index}
        className="text-center order-card shadow-lg"
        style={{ width: "18rem", height: "100%" }}
      >
        <Card.Header as="h5">Order #{outside.order[0].id_order}</Card.Header>
        <Card.Body>
          <Card.Subtitle className="mb-2 text-white order-by">
            By: {outside.order[0].username}
          </Card.Subtitle>
          {orderListInside}
          <div className="font-weight-bold">
            ${outside.order[0].order_total}
          </div>
        </Card.Body>
        <Card.Footer>
          <Form onSubmit={props.onSubmit}>
            <input
              name="id"
              defaultValue={outside.order[0].id_order}
              hidden
            />
            <Button type="submit" className="btn btn-dark">
              Mark as ready
            </Button>
          </Form>
        </Card.Footer>
      </Card>
    );
  }
}

```

```

        </Card.Footer>
      </Card>
    );
  });

  return <CardDeck className={"order-card-deck"}>{orderList}</CardDeck>;
} else {
  return <Alert variant="secondary">No orders to prepare</Alert>;
}
}

function OtherOrders(props) {
  const orders = props.orders;
  const orderList = orders.map((item, index) => {
    return (
      <tr key={index}>
        <td>{item.id_order}</td>
        <StatusTD
          status={item.status_order}
          onSubmit={props.onSubmit}
          id={item.id_order}
        ></StatusTD>
        <td>{item.order_hour}</td>
        <td>${item.order_total} MXN</td>
      </tr>
    );
  });

  return (
    <Table striped bordered hover size="sm" className="text-center shadow-lg">
      <thead>
        <tr>
          <th>Order #</th>
          <th>Status</th>
          <th>Hour</th>
          <th>Total price</th>
        </tr>
      </thead>
      <tbody>{orderList}</tbody>
    </Table>
  );
}

function StatusTD(props) {
  if (props.status !== undefined && props.status === "ready") {
    return (
      <React.Fragment>
        <td>
          {props.status}
          <Form size="sm" onSubmit={props.onSubmit}>
            <input name="id" defaultValue={props.id} hidden />
            <Button type="submit" variant="link">
              Mark as delivered
            </Button>
          </Form>
        </td>
      </React.Fragment>
    );
  }
}

```

```

    );
  } else if (props.status === "delivered") {
    return <td>{props.status}</td>;
  } else {
    return <td></td>;
  }
}
}

```

```
export default EmployeeDashboard;
```

## FoodMenu.js

```

import React, { Component } from "react";
import { Card, CardDeck } from "react-bootstrap";
import axios from "axios";

export class FoodMenu extends Component {
  state = {
    food: [],
    cart: [],
    totalprice: 0,
    index: 0,
    cartSubmitSuccess: false,
  };

  componentDidMount() {
    axios
      .get("/food/all")
      .then((res) => {
        this.setState({ food: res.data.data });
      })
      .catch((err) => console.log(err));
  }

  async getUserId() {
    let id = undefined;
    await axios
      .get("/user/credentials")
      .then((res) => {
        id = res.data.user;
      })
      .catch((err) => {
        console.log(err);
      });

    return id;
  }

  render() {
    return (
      <React.Fragment>
        <div className="gradient-line-secondary shadow-lg ">

```

```

    <h2 className="text-left p-3 text-uppercase">
      Login
      <br />
      to start <br />
      ordering <br />
      <span className="deli-text">delicious</span> <br />
      food today
    </h2>
  </div>
  <div className="container">
    <br />
    <h3>
      <span role="img" aria-label="blushing-smiley">
        😊
      </span>
      Today's menu
    </h3>
    <hr />
    <Food food={this.state.food} />
  </div>
</React.Fragment>
);
}
}

```

```

function Food(props) {
  const foods = props.food;
  const listFood = foods.map((food, index) => {
    return (
      <div key={index} className="col-auto mb-4">
        <Card
          className="text-justify shadow-lg"
          style={{ width: "18rem", height: "100%" }}
        >
          <Card.Header>{food.category}</Card.Header>
          <Card.Img
            variant="top"
            style={{
              height: "12vw",
              objectFit: "cover",
              width: "100%",
            }}
            className="food-photo"
            src={food.photo_url}
          />
          <Card.Body>
            <Card.Text>{food.name}</Card.Text>
            <Card.Text>
              Price: <b>${food.price}</b> MXN</b>
            </Card.Text>
          </Card.Body>
        </Card>
      </div>
    );
  });
};

return (
  <div className="row justify-content-center">
    <CardDeck>{listFood}</CardDeck>
  </div>

```

```
);
}
```

## Home.js

```
import React from "react";
import AwesomeSlider from "react-awesome-slider";
import "react-awesome-slider/dist/styles.css";
import { Card, Image } from "react-bootstrap";
import recycle from "../img/recycle.png";
import food from "../img/food.png";
import organic from "../img/organic.png";
import healthy from "../img/healthy.png";
import smoothie from "../img/smoothie.png";
import Button from "react-bootstrap/Button";
import { useHistory } from "react-router-dom";
import "../css/Home.css";
import axios from "axios";

function Home() {
  const history = useHistory();

  return (
    <React.Fragment>
      <h1 className="text-left home-title mt-4">
        Fast
        <br />
        Fruit
      </h1>
      <p className="home-subtitle">
        Eating healthy has never been more <span>delicious!</span>
      </p>

      <div className="gradient-line-main" />
      <AwesomeSlider
        className="mt-5 shadow-sm p-3 bg-white rounded"
        bullets={true}
      >
        <div data-src={require("../img/slider-1.jpg")} />
        <div data-src={require("../img/ffeat.jpg")} />
        <div data-src={require("../img/ffcups.jpg")} />
        <div data-src={require("../img/bg.jpg")} />
        <div data-src={require("../img/slider-7.jpg")} />
      </AwesomeSlider>
      <Description />
      <div className="jumbotron jumbotron-fluid know-us shadow-lg">
        <div className="container">
          <h2 className="text-center text-uppercase">
            What we offer in each food
          </h2>
        </div>
      </div>
      <div className="d-flex justify-content-center">
        <InfoCards />
      </div>
      <div className="d-flex justify-content-center">
```

```

    <Button
      className="btn-green shadow-lg"
      variant="primary"
      onClick={() => {
        navigateToFoodMenu(history);
      }}
    >
    Check out the menu!
  </Button>
</div>
</React.Fragment>
);
}

```

```

async function navigateToFoodMenu(history) {
  let user = await getUserCredentials();

  if (user.type === undefined) {
    history.push("/FoodMenu");
  } else {
    history.push("/client/order");
  }
}

```

```

async function getUserCredentials() {
  let type = undefined;
  let id = undefined;
  await axios
    .get("/user/credentials")
    .then((res) => {
      type = res.data.type;
      id = res.data.id;
    })
    .catch((err) => {
      console.log(err);
    });
}

```

```

const user = {
  id: id,
  type: type,
};
return user;
}

```

```

function Description() {
  return (
    <div className="row">
      <div className="col-lg-5 text-center">
        <Image src={smoothie} fluid />
      </div>
      <div className="col-lg-7 text-center info-panel">
        <h4>
          <b>Who are we?</b>
        </h4>
        <p>
          We are a concept of healthy fast food restaurants that adapt to the

```

needs of the modern lifestyle, offering our customers practical and economical options. We offer a menu with delicious, customizable food that is as healthy as the customer wants it to be, prepared on the spot with the freshest ingredients.

```

    </p>
  </div>
</div>
);
}

function InfoCards() {
  return (
    <div className="info-cards">
      <div className="row">
        <Card className="col-lg-3 col-md-5 col-xs-5 card-eco shadow-sm p-3 mb-5 bg-white rounded">
          <Card.Img variant="top" src={organic} className="card-img" />
          <Card.Body>
            <Card.Title>Eco-friendliness</Card.Title>
            <Card.Text>
              We encourage our customers to help the environment by giving them eco-friendly materials.
            </Card.Text>
          </Card.Body>
        </Card>
        <Card className="col-lg-3 col-md-5 col-xs-5 card-taste shadow-sm p-3 mb-5 bg-white rounded">
          <Card.Img variant="top" src={food} className="card-img" />
          <Card.Body>
            <Card.Title>Great Taste</Card.Title>
            <Card.Text>
              We delight our customers with delicious food, full of taste.
            </Card.Text>
          </Card.Body>
        </Card>
        <Card className="col-lg-3 col-md-5 col-xs-5 card-ingredients shadow-sm p-3 mb-5 bg-white rounded">
          <Card.Img variant="top" src={recycle} className="card-img" />
          <Card.Body>
            <Card.Title>Natural Ingredients</Card.Title>
            <Card.Text>
              Our goal is to give our customers the most delicious food made with natural ingredients.
            </Card.Text>
          </Card.Body>
        </Card>
        <Card className="col-lg-3 col-md-5 col-xs-5 card-healthy shadow-sm p-3 mb-5 bg-white rounded">
          <Card.Img variant="top" src={healthy} className="card-img" />
          <Card.Body>
            <Card.Title>Health</Card.Title>
            <Card.Text>
              Apart from our food being delicious and made with natural ingredients, it is also healthy.
            </Card.Text>
          </Card.Body>
        </Card>
      </div>
    </div>
  );
}

```

```
export default Home;
```

## Login.js

```
import React, { Component } from "react";
import { Alert } from "react-bootstrap";
import axios from "axios";

export class Login extends Component {
  constructor(props) {
    super(props);
    this.handleChange = this.handleChange.bind(this);
    this.handleSubmit = this.handleSubmit.bind(this);
  }

  state = {
    inputUsername: undefined,
    inputPassword: undefined,
    year: "-",
    error: undefined
  };

  async componentDidMount() {
    const currentDate = new Date();
    const currentYear = currentDate.getFullYear();
    this.setState({ year: currentYear });
  }

  async handleChange(event) {
    // This method lets us control multiple inputs in a form.
    // When an input in a form changes, this.state also updates
    // to form a single source of truth
    const target = event.target;
    const value = target.value;
    const name = target.name;

    await this.setState({
      [name]: value
    });
  }

  handleSubmit(event) {
    event.preventDefault();
    this.loginUser(this.state);
  }

  loginUser(userinfo) {
    let user = {
      inputUsername: userinfo.inputUsername,

```



```

    inputPassword: userinfo.inputPassword
  };

  axios({
    method: "post",
    url: "/loginUser",
    data: user
  })
  .then(res => {
    this.setState({ error: undefined });
    window.location.href = "/";
  })
  .catch(err => {
    this.setState({ error: "Incorrect username or password" });
  });
}

render() {
  return (
    <form className="form-signin" onSubmit={this.handleSubmit}>
      <div className="text-center mb-4">
        <h1 className="h3 mb-3 font-weight-normal">Login</h1>
        <p>
          You don't have an account? <a href="/register">Register</a>
        </p>
      </div>

      <div className="form-label-group">
        <label htmlFor="inputUsername">Username</label>
        <input
          id="inputUsername"
          name="inputUsername"
          className="form-control"
          placeholder="Username"
          required
          autoFocus=""
          autoComplete="off"
          onChange={this.handleInputChange}
        />
      </div>

      <div className="form-label-group">
        <label htmlFor="inputPassword">Password</label>
        <input
          type="password"
          id="inputPassword"
          name="inputPassword"
          className="form-control"
          placeholder="Password"
          required
          autoComplete="off"
          onChange={this.handleInputChange}
        />
      </div>
      <br />
      <Alert variant="info">
        <ul>

```

```

        <li>Master admin username: admin</li>
        <li>Master admin password: admin</li>
    </ul>
    </Alert>
    <Error error={this.state.error} />
    <button className="btn btn-lg btn-orange btn-block" type="submit"
id="sendCredentials">
        Login
    </button>
    <p className="mt-5 mb-3 text-muted text-center">© {this.state.year}</p>
</form>
    );
}
}

function Error(props) {
    if (props.error === undefined) {
        return <div></div>;
    }
    return <Alert variant="danger">{props.error}</Alert>;
}

```

## Navigation.js

```

import React, { Component } from "react";
import { NavLink } from "react-router-dom";
import { Form, Navbar, Nav, Button } from "react-bootstrap";
import axios from "axios";
export class Navigation extends Component {
    constructor(props) {
        super(props);
        this.handleSubmit = this.handleSubmit.bind(this);
    }

    state = {
        isLoggedIn: false,
        type: undefined,
    };

    componentDidMount() {
        axios
            .get("/user/credentials")
            .then((res) => {
                // Changes to type of user login
                this.setState({ isLoggedIn: true, type: res.data.type });
            })
            .catch((err) => {
                this.setState({ isLoggedIn: false });
            });
    }

    handleSubmit(event) {
        event.preventDefault();
    }
}

```

```

axios
  .get("/logout")
  .then((res) => {
    this.setState({ isLoggedIn: false });
    window.location.href = "/login";
  })
  .catch((err) => {
    console.log(err);
    this.setState({ isLoggedIn: true, error: "User couldn't logout" });
  });
}

render() {
  return (
    <>
      <Navbar bg="white" expand="lg">
        <Navbar.Brand className="mr-5">
          <p className="text-left brand-logo mt-4">
            Fast
            <br />
            Fruit
          </p>
        </Navbar.Brand>
        <Navbar.Toggle aria-controls="basic-navbar-nav" />
        <Navbar.Collapse id="basic-navbar-nav">
          <Nav className="mr-auto text-center">
            <NavLink className="d-inline nav-option home-link w-100" to="/">
              Home
            </NavLink>
            <OrderLink type={this.state.isLoggedIn}></OrderLink>
            <AdminLink type={this.state.type}></AdminLink>
            <EmployeeLink type={this.state.type}></EmployeeLink>
          </Nav>

          <div className="access-butons">
            <MenuButtons
              isLoggedIn={this.state.isLoggedIn}
              onSubmit={this.handleSubmit}
            />
          </div>
        </Navbar.Collapse>
      </Navbar>
    </>
  );
}

function GuestMenuButtons(props) {
  return (
    <React.Fragment>
      <NavLink
        variant="outline-primary"
        className="Nav.Link btn btn-orange shadow-lg"
        to="/login"
      >
        Login
    </React.Fragment>
  );
}

```

```

    </NavLink>
    <NavLink
      variant="outline-primary"
      className="Nav.Link btn btn-orange shadow-lg"
      to="/register"
    >
      Register
    </NavLink>
  </React.Fragment>
);
}

function UserMenuButtons(props) {
  return (
    <React.Fragment>
      <Form onSubmit={props.onSubmit}>
        <Button
          type="submit"
          value="Logout"
          className="btn btn-orange shadow-lg"
        >
          Logout
        </Button>
      </Form>
    </React.Fragment>
  );
}

function MenuButtons(props) {
  if (props.isLoggedIn) {
    return <UserMenuButtons onSubmit={props.onSubmit} />;
  } else {
    return <GuestMenuButtons onSubmit={props.onSubmit} />;
  }
}

function OrderLink(props) {
  if (props.type) {
    return (
      <NavLink
        className="d-inline nav-option order-link w-100"
        to="/client/order"
      >
        Order
      </NavLink>
    );
  } else {
    return (
      <NavLink className="d-inline nav-option menu-link w-100" to="/FoodMenu">
        Menu
      </NavLink>
    );
  }
}

function AdminLink(props) {
  if (

```

```

    props.type === undefined ||
    props.type === "client" ||
    props.type === "employee"
  ) {
    return null;
  } else if (props.type === "admin") {
    return (
      <NavLink
        className="d-inline nav-option admin-link w-100"
        to="/admin/tools"
      >
        Admin tools
      </NavLink>
    );
  } else {
    return null;
  }
}

```

```

function EmployeeLink(props) {
  if (props.type === undefined || props.type === "client") {
    return null;
  } else if (props.type === "admin" || props.type === "employee") {
    return (
      <NavLink
        className="d-inline nav-option employee-link w-100"
        to="/employee/tools"
      >
        Employee tools
      </NavLink>
    );
  } else {
    return null;
  }
}

```

## Order.js

```

import React, { useState, useEffect, useContext } from "react";
import { Card, Button, Alert, Form, CardDeck, Table } from "react-bootstrap";
import { OrderListContext } from "../context/Context";
import axios from "axios";
import { distance, preciseRound } from "../constants/Functions";
import "react-responsive-modal/styles.css";
import { Modal } from "react-responsive-modal";

```

```

function Order(props) {
  const notifyNewOrder = props.notifyNewOrder;
  const setNotifyNewOrder = props.setNotifyNewOrder;
  const [orderList, setOrderList] = useContext(OrderListContext);
  const [cart, setCart] = useState([]);
  const [foodList, setFoodList] = useState([]);
  const [storeList, setStoreList] = useState([]);
  const [selectedStore, setSelectedStore] = useState("");
  const [displayMap, setDisplayMap] = useState(false);
  const [modalState, setModalState] = useState(false);
  const [order, setOrder] = useState({

```

```

    totalprice: 0,
    index: 0,
    cartSubmitSuccess: false,
  });

useEffect(() => {
  setOrder({
    totalprice: 0,
    index: 0,
    cartSubmitSuccess: false,
    orderAccepted: false,
  });

  axios
    .get("/food/all")
    .then((res) => {
      setFoodList(res.data.data);
    })
    .catch((err) => console.log(err));
  var userCoordinates = [];
  //API to get user latitude and longitud
  axios("https://freegeoip.app/json/")
    .then((res) => {
      userCoordinates = [res.data.latitude, res.data.longitude];
      return userCoordinates;
      //find which store is closes to the user
    })
    .catch((err) => {
      //This usually fails if the user has adblock or some other
      //similar extension. Default value then is Disney's
      userCoordinates = [19.2853, -99.141301];
      console.log(err);
      return userCoordinates;
    })
    .then((userCoordinates) => {
      axios("/stores/all")
        .then((res) => {
          var orderedStores = [];
          res.data.data.forEach((store) => {
            let holder = distance(
              userCoordinates[0],
              userCoordinates[1],
              store.latitude,
              store.longitude,
              "g"
            );
            orderedStores.push({
              name: `${store.name}: (${store.location})`,
              distance: holder,
              latitude: store.latitude,
              longitude: store.longitude,
              id_store: store.id_store,
            });
          });
          orderedStores.sort((a, b) => {
            return a.distance - b.distance;
          });
          setStoreList(orderedStores);
        });
    });

```

```

        const url = `
https://embed.waze.com/iframe?zoom=16&lat=${orderedStores[0].latitude}&lon=${orderedStores[0].longitude}&ct=livemap`;

        setSelectedStore(url);
    })
    .catch((err) => {
        console.log(err);
    });
});
}, []);

const addToCart = async (event) => {
    event.preventDefault();
    event.persist();

    // Moves the user to the login page if he isn't logged in
    const loggedUserId = await getUserId();
    if (loggedUserId === undefined) {
        // Returns to login
        return (window.location.href = "/login");
    }

    const id = event.target.id.value; // Gets the input with the id
    const name = event.target.name.value; // Gets the input with the name
    const price = event.target.price.value; // Gets the input with the price
    const qty = event.target.qty.value; // Gets the input with the quantity
    const price_x_quantity = parseFloat(qty) * parseFloat(price);
    const price_x_quantity_rounded = preciseRound(price_x_quantity, 2);
    const new_cart = cart;
    new_cart.push({
        id_product: id,
        name: name,
        quantity: qty,
        price: price_x_quantity_rounded,
    });

    const oldTotalPrice = order.totalprice;
    const newTotalPrice =
        parseFloat(oldTotalPrice) + parseFloat(price_x_quantity);
    const roundedPriceTwoDecimals = preciseRound(newTotalPrice, 2);

    setOrder({
        cart: new_cart,
        totalprice: roundedPriceTwoDecimals,
    });

    // Reset the fields and load a quick animation
    event.target.qty.value = 0;
    event.target.submit_btn.innerHTML = "&#10003;"; // Check mark
    event.target.submit_btn.classList.add("check");
    setTimeout(function () {
        event.target.submit_btn.innerHTML = "+";
        event.target.submit_btn.classList.remove("check");
    }, 1000);
};

```

```

    }, 1500);
  };

  const confirmedOrder = (event) => {
    setDisplayMap(true);
  };

  const cartSubmit = async (event) => {
    event.preventDefault();
    event.persist();
    // Moves the user to the login page if he isn't logged in
    const loggedInUserId = await getUserId();
    if (loggedInUserId === undefined) {
      // Returns to login
      return (window.location.href = "/login");
    }

    const body = {
      id: loggedInUserId,
      cart: cart,
    };

    axios({
      method: "post",
      url: "/order",
      data: body,
    })
      .then((res) => {
        setCart([]);
        setOrder({ cartSubmitSuccess: true, totalprice: 0, index: 0 });
        setNotifyNewOrder(!notifyNewOrder);
        setOrderList(!orderList);
        setDisplayMap(false);
        setModalState(true);
      })
      .catch((err) => {
        setOrder({ cartSubmitSuccess: false });
        console.log(err);
      });
  };

  const changeMap = (event) => {
    const target = event.target;
    const value = target.value;
    const name = target.name;
    console.log(value, name);
    storeList.forEach((store) => {
      if (store.id_store === value) {
        setSelectedStore(
          `https://embed.waze.com/iframe?zoom=16&lat=${store.latitude}&lon=${store.longitu
de}&ct=livemap`
        );
      }
    });
  };
};

```



```

const getUserId = async () => {
  let id = undefined;
  await axios
    .get("/user/credentials")
    .then((res) => {
      id = res.data.user;
    })
    .catch((err) => {
      console.log(err);
    });

  return id;
};

const closeModal = () => {
  setModalState(false);
};

return (
  <React.Fragment>
    <Modal open={modalState} onClose={closeModal}>
      <h5>
        <strong>Your order has been submitted</strong>
        <span role="img" aria-label="blushing-smiley">
          😊
        </span>
      </h5>
      <p>
        You'll receive a notification when your order is ready for pick-up!
      </p>
    </Modal>
    <div className="gradient-line-secondary shadow-lg">
      <h2 className="text-left p-3 text-uppercase">
        Order <br />
        something <br />
        <span className="deli-text">delicious</span> <br />
        today
      </h2>
    </div>
    <div className="container">
      <br />
      <h3>
        <span role="img" aria-label="shopping cart">
          🛒
        </span>{ " "}
        Cart
      </h3>
      <hr />
      <Cart
        totalprice={order.totalprice}
        cart={cart}
        cartSubmitSuccess={order.cartSubmitSuccess}
        stores={storeList}
        selected={selectedStore}
        handleChange={changeMap}
        onSubmit={confirmedOrder}
        sendOrder={cartSubmit}
        displayMap={displayMap}
      />
    </div>
  </React.Fragment>
);

```

```

    />
    <br />
    <h3>
      <span role="img" aria-label="salad">
        
      </span>
      Food
    </h3>
    <hr />
    <Food food={foodList} onSubmit={addToCart} />
  </div>
</React.Fragment>
);
}

function Food(props) {
  const foodList = props.food;
  const listFood = foodList.map((food, index) => {
    return (
      <div key={index} className="col-auto mb-4">
        <Card
          className="text-justify shadow-lg"
          style={{ width: "18rem", height: "100%" }}
        >
          { /* <div className="card-overlay"></div> */ }
          <Card.Header className="text-center">
            <b>{food.category}</b>
          </Card.Header>
          <Card.Img
            variant="top"
            style={{
              height: "12vw",
              objectFit: "cover",
              width: "100%",
            }}
            className="food-photo"
            src={food.photo_url}
          />
          <Card.Body>
            <Card.Text>{food.name}</Card.Text>
            <Card.Text>
              Price: <strong>${food.price} MXN</strong>
            </Card.Text>
          </Card.Body>
          <Card.Footer className="text-muted">
            <Form onSubmit={props.onSubmit}>
              <div className="form-group food-form-group ">
                <input name="id" defaultValue={food.id_product} hidden />
                <input name="name" defaultValue={food.name} hidden />
                <input name="price" defaultValue={food.price} hidden />
                <input
                  type="number"
                  id= {"qty"+index}
                  name="qty"
                  className="form-control"
                  placeholder="Qty."
                  style={{ height: "100%" }}
                  min="1"
                  max="5"
                  required
                />
              </div>
            </Form>
          </Card.Footer>
        </div>
      );
    );
  });
}

```

```

        <Button
          type="submit"
          name="submit_btn"
          className="btn-sm btn-green btn-add-order shadow-sm"
          id={"addItem"+index}
        >
        <strong>+</strong>
      </Button>
    </div>
  </Form>
</Card.Footer>
</Card>
</div>
);
});

return (
  <div className="row justify-content-center">
    <CardDeck>{listFood}</CardDeck>
  </div>
);
}

```

```

function Cart(props) {
  const cart = props.cart;
  const url = String(props.selected);
  if (cart.length > 0) {
    return (
      <>
        <label> The order you are placing has: </label>

        <CartListItems
          totalprice={props.totalprice}
          cart={props.cart}
          onSubmit={props.onSubmit}
        />
        {props.displayMap} && (
          <>
            <div className="text-center shadow-lg">
              <select
                id="inputStore"
                name="inputStore"
                className="form-control"
                required
                onChange={props.handleChange}
              >
                <DeliveryList stores={props.stores} />
              </select>
            <div>
              <iframe
                width="100%"
                height="300px"
                frameborder="0"
                scrolling="no"
                marginheight="0"
                marginwidth="0"
                src={url}
                title="Mapa waze"
              >>/iframe>
            </div>
          </>
        )
      </>
    );
  }
}

```

```

        </div>
        <Form onSubmit={props.sendOrder}>
          <Button
            type="submit"
            className="btn-block btn-green"
            marginLeft="50%"
          >
            <span className=" blink-anim">SUBMIT ORDER</span>
          </Button>
        </Form>
      </div>
    </>
  )}
</>
);
} else {
  return (
    <React.Fragment>
      <Alert variant="secondary">
        No items in your cart. Let's order something!
      </Alert>
    </React.Fragment>
  );
}
}

function CartListItems(props) {
  const cartListItems = props.cart.map((item, index) => {
    return (
      <tr key={index}>
        <td>{item.name}</td>
        <td>x{item.quantity}</td>
        <td>${item.price} MXN</td>
      </tr>
    );
  });
  cartListItems.push(
    <React.Fragment>
      <tr className="font-weight-bold">
        <td colspan="2" className="text-center">
          Total price
        </td>
        <td>-- ${props.totalprice} MXN --</td>
      </tr>
      <tr>
        <td colspan="3">
          <Button type="submit" variant="link" onClick={props.onSubmit}>
            Confirm order
          </Button>
        </td>
      </tr>
    </React.Fragment>
  );
  return (
    <Table striped bordered hover size="sm" className="text-center shadow-lg">
      <thead>
        <tr>
          <th>Name</th>
          <th>Quantity</th>
          <th>Price</th>
        </tr>

```

```

        </thead>
        <tbody>{cartListItems}</tbody>
    </Table>
    );
}

function DeliveryList(props) {
    const stores = props.stores;
    const listStores = stores.map((value, index) => {
        return (
            <option key={value.id_store} value={value.id_store}>
                {value.name} is {`${Math.floor(value.distance * 1000)}m away`}
            </option>
        );
    });

    return <React.Fragment>{listStores}</React.Fragment>;
}

export default Order;

```

## Register.js

```

import React, { Component } from "react";
import { Alert, Form } from "react-bootstrap";
import axios from "axios";

export class Register extends Component {
    constructor(props) {
        super(props);
        this.handleChange = this.handleChange.bind(this);
        this.handleSubmit = this.handleSubmit.bind(this);
    }

    state = {
        inputUsername: undefined,
        inputType: "client",
        inputPassword: undefined,
        inputConfirmPassword: undefined,
        year: "_",
        errors: [],
        success: undefined
    };

    async componentDidMount() {
        const currentDate = new Date();
        const currentYear = currentDate.getFullYear();
        this.setState({ year: currentYear });
    }

    async handleChange(event) {

```

```

// This method lets us control multiple inputs in a form.
// When an input in a form changes, this.state also updates
// to form a single source of truth
const target = event.target;
const value = target.value;
const name = target.name;

await this.setState({
  [name]: value
});
}

handleSubmit(event) {
  event.preventDefault();
  this.createUser(this.state);
}

createUser(userinfo) {
  let user = {
    inputUsername: userinfo.inputUsername,
    inputType: userinfo.inputType,
    inputPassword: userinfo.inputPassword,
    inputConfirmPassword: userinfo.inputConfirmPassword
  };

  axios({
    method: "post",
    url: "/registerUser",
    data: user
  })
  .then(res => {
    if (res.data.status !== undefined) {
      this.setState({
        success: undefined,
        errors: res.data.message.errors
      });
    } else {
      this.setState({ success: res.data.message, errors: [] });
    }
  })
  .catch(err => {
    console.log(err);
    this.setState({ success: undefined, errors: err.description });
    console.log(err);
  });
}

render() {
  return (
    <div className="container-fluid">
      <Form className="form-signin" onSubmit={this.handleSubmit}>
        <div className="text-center mb-4">
          <h1 className="h3 mb-3 font-weight-normal">Register</h1>
          <p>
            Already have an account? <a href="/login">Login</a>
          </p>
        </div>
      </Form>
    </div>
  );
}

```

```

    </div>
    <div className="form-label-group">
      <label htmlFor="inputUsername">Username</label>
      <input
        id="inputUsername"
        name="inputUsername"
        className="form-control"
        value={this.state.username}
        onChange={this.handleInputChange}
        required
      />
    </div>
    <div className="form-label-group">
      <label htmlFor="inputPassword">Password</label>
      <input
        type="password"
        id="inputPassword"
        name="inputPassword"
        className="form-control"
        value={this.state.password}
        onChange={this.handleInputChange}
        required
      />
    </div>
    <div className="form-label-group">
      <label htmlFor="inputConfirmPassword">Confirm password</label>
      <input
        type="password"
        id="inputConfirmPassword"
        name="inputConfirmPassword"
        className="form-control"
        value={this.state.hashdpassword}
        onChange={this.handleInputChange}
        required
      />
    </div>
    <br />
    <Success success={this.state.success} />
    <Errors errors={this.state.errors} />
    <br />
    <button className="btn btn-lg btn-orange btn-block" type="submit">
      Register
    </button>
    <p className="mt-5 mb-3 text-muted text-center">
      © {this.state.year}
    </p>
  </Form>
</div>
);
}
}

function Success(props) {
  if (props.success !== undefined) {
    return <Alert variant="success">User created successfully</Alert>;
  } else {
    return <div></div>;
  }
}
}

```

```
function Errors(props) {
  if (!props.errors || props.errors.length === 0) {
    return <div></div>;
  }
  return (
    <Alert variant="danger">
      Oops! the following errors were found:
      <ul>
        {props.errors.map((value, index) => {
          return <li key={index}>{value.msg}</li>;
        })}
      </ul>
    </Alert>
  );
}
```

## package.json

```
{
  "name": "restaurant",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^4.2.4",
    "@testing-library/react": "^9.5.0",
    "@testing-library/user-event": "^7.2.1",
    "axios": "^0.19.2",
    "bootstrap": "^4.4.1",
    "react": "^16.13.1",
    "react-awesome-slider": "^4.1.0",
    "react-bootstrap": "^1.0.0-beta.17",
    "react-dom": "^16.13.1",
    "react-responsive-modal": "^5.0.2",
    "react-router-dom": "^5.1.2",
    "react-scripts": "3.4.1",
    "react-toastify": "^5.5.0",
    "socket.io-client": "^2.3.0"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": "react-app"
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}
```



```

    },
    "proxy": "http://localhost:5000"
  }
}

```

## app.js

```

// configs/app.js
const dotenv = require("dotenv");

dotenv.config();

const appConfig = {
  env: process.env.APP_ENV || "development",
  expressPort: process.env.EXPRESS_PORT || 3306,
  secret: process.env.APP_SECRET || "YOU_SHOULD_NOT_USE_THIS_SECRET"
};

module.exports = appConfig;

```

## passport.js

```

let passport = require("passport");
let LocalStrategy = require("passport-local").Strategy;
let UserModel = require("../models/User");
let bcrypt = require("bcrypt");

const userTableFields = {
  usernameField: "inputUsername",
  passwordField: "inputPassword"
};

const verifyCallback = async (inputUsername, inputPassword, done) => {
  await UserModel.findByUsername(inputUsername)
    .then(user => {
      if (!user || user.length == 0) {
        return done(null, false);
      }

      let isValid = bcrypt.compareSync(inputPassword, user[0].password);
      if (isValid) {
        return done(null, user[0]);
      } else {
        return done(null, false);
      }
    })
    .catch(err => {
      done(err);
    });
});

```

```
const strategy = new LocalStrategy(userTableFields, verifyCallback);
```

```
passport.use(strategy);
```

```
passport.serializeUser((user, done) => {
  done(null, user.username);
});
```

```
passport.deserializeUser((id, done) => {
  UserModel.findByUsername(id)
    .then(user => {
      done(null, user[0]);
    })
    .catch(err => done(err));
});
```

## AuthController.js

```
const { validationResult } = require("express-validator");
let UserModel = require("../models/User");
const bcrypt = require("bcrypt");
```

```
exports.registerUser = (req, res) => {
  const errors = validationResult(req);
  if (!errors.isEmpty()) {
    return res.json({
      status: 500,
      message: errors,
    });
  }
}
```

```
let username = req.body.inputUsername;
let type = req.body.inputType;
let password = req.body.inputPassword;
let hashedPass = bcrypt.hashSync(password, 10);
let id_store;
```

```
// Nulls the id_store if the user is a client type, since it's not related to a specific store
```

```
if (type === "client" || req.body.inputStore === undefined) {
  id_store = undefined;
} else {
  id_store = req.body.inputStore;
}
```

```
UserModel.createUser(username, type, hashedPass, id_store)
  .then((_) => {
    res.json({ message: "User created succesfully" });
  })
  .catch((error) => {
    console.log(error);
  });
```

```

    res.status(500).json({ message: error });
  });
};

exports.userCredentials = (req, res) => {
  if (req.user !== undefined) {
    return res.json({
      user: req.user.username,
      type: req.user.type,
      id: req.user.id_user,
    });
  } else {
    return res.status(500).json({ message: "No logged in user" });
  }
};

exports.logout = (req, res) => {
  req.logout();
  res.json({ message: "Logout successfull" });
};

```

## FoodController.js

```

let FoodModel = require("../models/Food");

exports.allFood = (req, res) => {
  FoodModel.allProducts()
    .then(data => {
      res.json({ data: data });
    })
    .catch(error => {
      res.status(500).json({ message: error });
    });
};

```

## StoreController.js

```

let StoreModel = require("../models/Store");

exports.getAllStores = (req, res) => {
  return StoreModel.findAll()
    .then(data => {
      res.json({ data: data });
    })
    .catch(error => {
      res.status(500).json({ message: error });
    });
};

exports.getAllEmployees = (req, res) => {
  return StoreModel.findAllEmployees()
    .then(data => {
      res.json({ data: data });
    });
};

```

```

    })
    .catch(error => {
      res.status(500).json({ message: error });
    });
  });
};

```

## connection.js

```

let appConfig = require("../configs/app");
const knexfile = require("../knexfile");
const knex = require("knex")(knexfile[appConfig.env]);
module.exports = knex;

```

## routes.js

```

var express = require("express");
var router = express.Router();
let authValidator = require("../validators/AuthValidator");
let passport = require("passport");
/* Controllers */
let authController = require("../controllers/AuthController");
let foodController = require("../controllers/FoodController");
let orderController = require("../controllers/OrderController");
let storeController = require("../controllers/StoreController");

/* Routes */
router.post(
  "/registerUser",
  authValidator.registerUser,
  authController.registerUser
);

router.post(
  "/loginUser",
  passport.authenticate("local", {
    successRedirect: "/",
    failureRedirect: "/register",
  })
);

router.get("/user/credentials", authController.userCredentials);

router.get("/logout", authController.logout);

router.get("/food/all", foodController.allFood);

router.post("/order", orderController.createOrder);

router.get("/order/all", orderController.showAllOrders); //show all orders

```

```

router.get("/order/all/:id", orderController.showAllOrdersFromId); //show all orders from
a specific user_id

router.get("/order/readyOrDelivered", orderController.findRDOrders);

router.post("/order/ready/:id", orderController.orderReady); //mark order as ready

router.post("/order/delivered/:id", orderController.orderDelivered); //mark order as
delivered

router.get("/order/products/:id", orderController.productsFromOrder); //get products from
certain order

router.get("/order/list", orderController.showUsersOrders); //show a user's orders

router.get("/order/preparing", orderController.getPreparingOrdersIds);

router.get("/stores/all", storeController.getAllStores);

router.get("/employees/all", storeController.getAllEmployees);

module.exports = router;

```

## knexfile.js

```

const dotenv = require("dotenv");

dotenv.config();

module.exports = {
  development: {
    client: "mysql2",
    connection: {
      host: process.env.DB_DEVELOPMENT_HOST || "localhost",
      port: process.env.DB_DEVELOPMENT_PORT || "3306",
      database: process.env.DB_DEVELOPMENT_NAME || "restaurant",
      user: process.env.DB_DEVELOPMENT_USER || "root",
      password: process.env.DB_DEVELOPMENT_PASSWORD || ""
    },
    pool: {
      min: 2,
      max: 10
    },
    migrations: {
      tableName: "knex_migrations"
    }
  }
}

```

```
  },
  production: {
    client: "mysql2",
    connection: {
      host: process.env.DB_PRODUCTION_HOST || "localhost",
      port: process.env.DB_PRODUCTION_PORT || "3306",
      database: process.env.DB_PRODUCTION_NAME || "restaurant",
      user: process.env.DB_PRODUCTION_USER || "root",
      password: process.env.DB_PRODUCTION_PASSWORD || ""
    },
    pool: {
      min: 2,
      max: 10
    },
    migrations: {
      tableName: "knex_migrations"
    }
  }
};
```