

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Кафедра електроніки і комп'ютерної техніки

ПОЯСНЮВАЛЬНА ЗАПИСКА

ДО КВАЛІФІКАЦІЙНОЇ
РОБОТИ БАКАЛАВРА

на тему:

«Програмно-апаратний комплекс документування даних
лікарняних закладів»

Завідувач кафедри ЕКТ

А.С. Опанасюк

Керівник роботи

І.А. Кулик

Студент групи ЕС-61

А.О. Хацько

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Факультет _____ електроніки та інформаційних технологій
Кафедра _____ електроніки і комп'ютерної техніки
Напрямок підготовки _____ 171 Електроніка
Освітня програма _____ Електронні системи та компоненти

ЗАТВЕРДЖУЮ

Зав. кафедрою _____ Опанасюк А.С..

"___" _____ 202__ р..

ЗАВДАННЯ

на кваліфікаційну роботу бакалавра

1 Тема роботи _____

затверджена наказом по університету "___" _____ 202__ р. № _____

2 Термін здачі студентом закінченої роботи _____

3 Вихідні дані до роботи _____

4 Зміст розрахунково-пояснювальної записки (перелік питань, що належить розробити): 1. Огляд літератури та поставка задачі. 2. Розроблення алгоритму роботи та структурної схеми програмно-апаратного комплексу. 3. Розробка принципової схеми мікропроцесорного пристрою контролю у складі програмно-апаратного комплексу. 4. Розробка загального програмного забезпечення для програмно-апаратного комплексу.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____

6 Дата видачі завдання _____

Керівник _____

Завдання прийняв до виконання _____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного проєкту	Термін виконання етапів роботи	Примітки
1	Аналіз існуючих аналогів	27.03.20 – 05.04.20	
2	Вибір засобів реалізації програмно-апаратного комплексу	05.04.20 – 11.04.20	
3	Розробка алгоритмів роботи комплексу	12.04.20 – 20.04.20	
4	Розробка структурної схеми	21.04.20 – 28.04.20	
5	Проектування підсистем обробки переривань та виводу на індикатор	29.04.20 – 14.05.20	
6	Розробка програмного забезпечення	15.05.20 – 25.05.20	
7	Створення пояснень до комплексу	26.05.20 – 01.06.20	
8	Оформлення пояснювальної записки й креслень	02.06.20 – 10.06.20	
9	Представлення роботи керівнику і отримання відзиву	11.06.20	
10	Представлення роботи кафедрі для отримання рецензії	14.06.20	

Студент _____

Керівник _____

" ____ " _____ 202__р

РЕФЕРАТ

Пояснювальна записка: 71 аркуш; 35 рисунків; 9 джерел.

Графічні матеріали: 1 принципова електрична схема.

Ключові слова: програмно-апаратний комплекс, мікропроцесорна система, мікропроцесорний пристрій, Intel 8259, документування даних, обробка інформації, створення меню.

Пояснювальна записка складається зі вступу, 4 розділів, висновку, списку літератури, 3 додатків та одного креслення.

Перший розділ включає огляд джерел, постановку завдання та розглядаються комплекси аналогів.

Другий розділ містить інформацію про розробку алгоритму роботи та структурної схеми програмно-апаратного комплексу.

Третій комплекс включає в себе розробку принципової схеми мікропроцесорного пристрою контролю у складі програмно-апаратного комплексу. Проведено вибір елементарної бази, розроблено підсистеми обробки переривань та виводу на індикатор, проведено розробку програмного забезпечення для роботи мікропроцесорного пристрою.

Четвертий розділ містить розробку загального програмного забезпечення комплексу, розроблено програму складання та обробки меню лікарняного закладу.

ЗМІСТ

Вступ.....	3
1 Огляд літератури та постановка задачі	4
1.1 Аналіз програмно-апаратних продуктів аналогів.....	4
1.2 Постановка задачі проєктування	9
2 Розроблення алгоритму роботи та структурної схеми програмно-апаратного комплексу.....	12
2.1 Синтез алгоритму функціонування програмно-апаратного комплексу	12
2.2 Синтез структурної схеми мікропроцесорної системи у складі програмно-апаратного комплексу.....	14
3 Розробка принципової схеми мікропроцесорного пристрою контролю у складі програмно-апаратного комплексу	17
3.1 Вибір елементної бази мікропроцесорної системи	17
3.2 Розробка підсистема обробки переривань.....	18
3.3 Розробка підсистеми виводу на матричний індикатор	19
3.4 Розробка програмного забезпечення для мікропроцесорної системи	22
4 Розробка загального програмного забезпечення для програмно-апаратного комплексу.....	23
4.1 Проєктування загального програмного забезпечення.....	23
4.2 Пояснення до програми	25
4.3 Розробка програмної частини програмно-апаратного комплексу ..	35

					ЕЛІТ 6.171.00.10.415 ПЗ			
Зм.	Лист	№ докум.	Підпис	Дата	Програмно-апаратний комплекс документування даних лікарняних закладів	Літ.	Аркуш	Аркушів
Розроб.		Хацько А. О.					1	71
Перевір.								
Т. Контр.								
Н. Контр.								
Затверд.		Опанасюк А.С.			СумДУ, зр. ЕС-61			

Висновки	45
Список літератури.....	46
Додаток А.....	47
Додаток Б	54
Додаток В.....	55

					ЕЛІТ 6.171.00.10.415 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		2

ВСТУП

Ми живемо у світі, який сколихнув бум розвитку високих технологій. Інноваційні системи використовуються в більшості розвинених країнах. В час, коли людський ресурс намагаються використовувати максимально оптимізовано, витиснути найбільший коефіцієнт корисної дії за робочий час підприємства. Для цього велика кількість компаній та підприємств впроваджують апаратно-програмні комплекси, котрі спрощують та прискорюють виконання технічного процесу, покращують взаємодію між різними підрозділами та відділами підприємств.

Наприклад, автовиробник застосовують комплекси автоматичного фарбуванням елементів автомобіля, комплекси автоматичної збірки кузовів авто. Кондитерські підприємства налаштовують конвеєрні ленти, на яких за допомогою запрограмованої логіки формуються великі партії виробів, котрі відповідають техлогічним вимогам. Лікарні в Україні працюють з комп'ютерами яким 15, іноді 20 років, потужності яких вже не вистачає для обробки всієї інформації, що на них надходить.

Нажаль, в нашій країні не все так добре з впровадженням технологій, особливо в державних установах. Це відбувається з декількох причин:

- загальна фінансова ситуація, коли грошей не вистачає навіть на забезпечення установ най необхіднішим, що вже говорити про фінансування розвитку;
- люди звикли працювати так, як їх колись навчили. І вони не бажають розвивати свої професійні навички та вчитися чомусь новому;
- мала ініціативність керівників та власників підприємств в впровадженні новітніх технологій на власні потужності.

Мета цієї роботи, розробка програмно-апаратного комплексу документування даних лікарняних закладів. Який дасть змогу поліпшити роботу цілого відділу лікарні, оптимізувавши зручність роботи та пришвидшити її виконання.

Даний комплекс складається с спеціально розробленого, під нього, програмного забезпечення на мові C#, з інтеграцією баз даних та робочого місця з персональним комп'ютером та промисловими засобами виводу інформації в друкованих вигляд.

					ЕЛІТ 6.171.00.10.415 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		3

1 ОГЛЯД ЛІТЕРАТУРИ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз програмно-апаратних продуктів аналогів

В наш час мабуть кожна лікарня має в своєму розпорядженні технічні засоби та програмне забезпечення, які спрощують функціонування медичного закладу. Якщо ж необхідна офісне техніка закуповується в ході тендеру, інформація про який знаходяться в відкритому доступі. То спеціалізоване програмне забезпечення розробляється найчастіше приватними компаніями або взагалі фрілансерами, які намагаються максимально зберегти свої продукти не поширюючи інформацію про них в мережі інтернет, для упередження крадіжки розробок з боку конкурентів.

Розглянемо проєкт документування даних «PITANIE.EXE», який було розроблено та впроваджено для потреб однієї з сумських лікарень. Автор розробки невідомий, відповідної технічної документації по цьому проєкту у відкритому доступі не знайдено. Апаратна частина реалізована на потужностях застарілих на 15-20 років, потужності і швидкодії яких не вистачає для зручного користування цією розробкою. Апаратна частина складається з:

1. Персонального комп'ютера.
2. Засобу виводу інформації - 19 дюймового монітора з роздільною здатністю HD 1366x768.
3. Засобу виводу інформації - принтера EPSON FX-1170.
4. Комп'ютерної миші та клавіатури.

Персональний комп'ютер має такі характеристики:

1. Процесор невідомого виробника з одним ядром та тактовою частотою 1.5 ГГц;
2. Відомості про відеокарту відсутні;
3. Оперативна пам'ять 2 Гб;
4. Жорсткий диск 120 Гб.

Комп'ютер працює під операційною системою Windows XP. Автор розробки виконав свій проєкт на мові програмування MS DOS. Програма керується за допомогою клавіатури, підтримка миші відсутня. Через це зручність користування під сумнівом. Графічна реалізація програми зображено на рисунку 1.1.1.

					ЕЛІТ 6.171.00.10.415 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		4

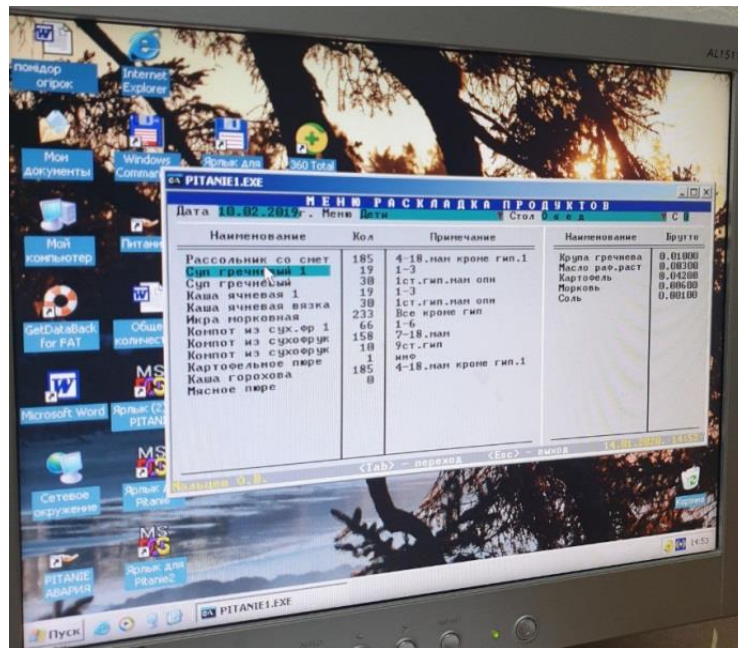


Рисунок 1.1.1 – Вигляд головного меню програмного забезпечення РІТАНІЕ1.ЕХЕ

Основні функції програми полягають в складанні меню харчування пацієнтів лікарні на один день. В програму вносять готові страви з технологічних карт, які розроблені кухарем-технологом. В програмі з вже внесених страв збирають 5 прийомів їжі: сніданок, пізній сніданок, обід, полуденок та вечеря. Після виконання всіх операцій з програми відправляється на друк готове меню (рис. 1.1.2).

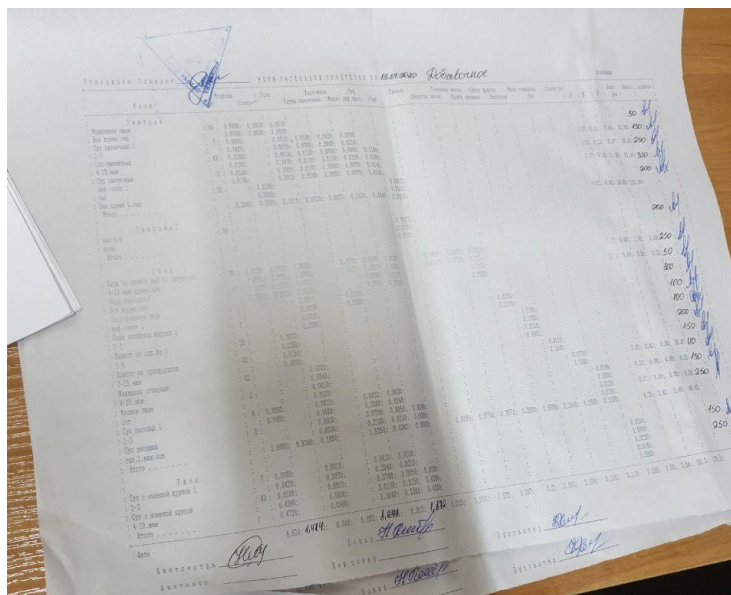


Рисунок 1.1.2 – Кінцевий результат роботи розглядаємої програми

Ця програма має велику кількість недоліків, основні з яких це:

1. Відсутність підтримки комп'ютерної миші, керування відбувається тільки клавіатурою;
2. Низька швидкість роботи додатку, через використання застарілих мов програмування;
3. Недосконалість введення інформації, під час кожного використання необхідно заново вводити назви страв та їх склад.

Також розглянемо програму від компанії «БЕСТ», під назвою «БЕСТ.ПИТАНИЕ» [1]. За заявою розробника це спеціальне рішення для закладів харчування, яке дозволяє розрахувати та замовити необхідну кількість продуктів, що дозволяє покращити та спростити роботу по складанню та розрахунку меню.

До основних можливостей цього програмного комплексу відносять можливість формувати меню відповідно до:

1. Діет, для лікарень, закладів реабілітації;
2. Замовлень, для закладів громадського харчування;
3. Збалансування по енергетичній цінності, складу мікроелементів, вітамінів та хімічному складі.

На рисунку 1.1.3 скріншот програми, під час розробки технологічної карти:

Номер	Наименование блюда	Вес	Хим. состав	
000127	Тыква отварная	110.0	Калорийность	69
000128	Фасоль овощная отварная	80.0	Белки.....	2.30
000129	Овощи, припущенные в молочном соусе	125.0	Жиры.....	2.60
000130	Овощи и фрукты тушеные	175.0	Углеводы....	9.00
000131	Капуста тушеная	125.0	Зола.....	0.00
000132	Картофель, тушеный с луком	125.0	Мин. вещества	
000133	Картофель тушеный	100.0	Натрий.....	0.00
000134	Картофель и овощи, тушеные с соусе	130.0	Калий.....	0.00
			Кальций.....	54.00
			Магний.....	26.00
			Фосфор.....	62.00
			Железо.....	1.00
			Цинк.....	0.00
			Йод.....	0.00
			Витамины	
			А.....	0.90
			В1.....	0.05
			В2.....	0.00
			В6.....	0.00
			РР.....	0.00
			С.....	8.00
			Е.....	0.40

N	N/номер	Наименование	Брутто	Нетто
0	0000000000001	Морковь	58.000	46.000
0	0000000000021	Репа	26.000	19.000
0	0000000000004	Горошек зеленый консе	16.000	10.000
0	0000000000005	Капуста белокочанная	25.000	20.000
0		Масса припущенных ово		85.000
0	0000000000001	Сахар	1.000	1.000
0	000366	Соус молочный (для по	40.000	40.000

Рисунок 1.1.3 – Вид технологической карты в программе «БЕСТ-5»

Також програма дає можливість розробки типового меню на день, на тиждень та на місяць. Наявна можливість використання меню-раскладки, яка розробляється кожного дня, для кожного прийому їжі. Це меню можна збирати як вручну, так і автоматично, на основі типового меню. Також наявна функція обліку продуктів, з можливістю фіксувати прихід та витрати продуктів, відстежувати їх переміщення. Ця інформація відстежується через створені меню.

Є можливість слідкувати за раціоном харчування за хімічним складом, енергетичною цінністю, що підтверджується рисунком 1.1.4, на якому зображено розрахунок калорійності харчування відповідно до прийомів їжі:

БЕСТ-5 /3.4 SP 24/ [Питание.Калькуляция.Питание организованных коллективов]

15/09/2010 12:45 | Начало расчетного периода: 01/01/2010

Кухня

РЕЕСТР МЕНЮ-РАСКЛАДОК

Дата	Прием	Категория	Диета	Кол	ДЗ	ВЗ	Р	Н
14/01/10	2-ой завтрак	Ясли		14			✓	
14/01/10	Обед	Ясли		14			✓	
14/01/10	Полдник	Ясли		14			✓	
14/01/10	Ужин							
14/01/10	Завтрак							
14/01/10	Обед							
14/01/10	Полдник							
14/01/10	Ужин							

Расчет калорийности на 14/01/10

Ясли (без диеты)

Прием	ККал	Белки	Жиры	Углеводы	с
Завтрак.....	333	12.71	15.79	34.90	0.0
2-ой завтрак..	81	0.71	0.00	19.45	5.0
Обед.....	575	23.76	14.72	86.96	0.0
Полдник.....	185	6.18	4.50	29.86	0.0
Ужин.....	274	7.59	8.49	41.71	2.0
Итого за день:	1448	50.95	43.50	212.88	2.0
Норма - min:	1540	53.00	53.00	212.00	2.0
Норма - max:	1540	53.00	53.00	212.00	2.0

Требование: сформировано

№	Код		с
*	1	БРЩ1/2 Борщ из све	0.0
*	2	Мак1 Запеканка в	5.0
*	2	огурец1 Огурцы порц	0.0
*	3	10/1 Компот из с	0.0
*	4	БУТ0003 Хлеб ржаной	0.0
*	5	б/н Укроп	2.0
*	5	б/н Лук зеленый	2.0
*	5	Соль1 Соль на весь день	14 2.0

ГЛАВНОЕ МЕНЮ | Питание.Калькуляция

DENIS admin Летский сад 339(2010) (C:\Program Files\BEST5\BEST5_34\Server\Date\DB\FD\339_10\)

Рисунок 1.1.4 – Калькуляція калорійності харчування відповідно до прийомів їжі

Як висновок, роглянуті вище програми в деякому сенсі виконують схожі функції розробляемого в цій роботі апаратно-програмного комплексу, але вони мають суттєві недоліки. Програма «PITANIE.EXE» складна та не зручна в використанні, обмежена в функціонуванні, розроблена на застарілій мові програмування, не має підтримки від розробника, в ході її роботи виникають критичні помилки, які призводять до аварійного зупинення програми без збереження інформації. Програма «БЕСТ.ПИТАНИЕ» має ширший функціонал, зручніша в використанні, позбавлена критичних помилок. Але вона також застаріла, через що підтримка цього продукту під сумнівом та через її комерційну цінність в вільному доступі вона відсутня, а доступна лише після покупки у компанії виробника.

1.2 Постановка задачі проєктування

Метою даної кваліфікаційної роботи бакалавра є розробка програмно-апаратного комплексу документування даних для їдальні лікарні, в якому буде зручно та швидко складати меню на день для пацієнтів.

Комплекс повинен відповідати таким вимогам та мати:

1. Швидкодію складання та обробки меню менше 10 мс.
2. Зручність та простоту користування, для використання персоналом з недостатніми знаннями ПК.
3. Використання актуальних технологій та сучасної операційної системи на комп'ютері, на якому буде працювати програмна частина комплексу.
4. Розроблення, складання та підрахунок прийомів їжі та страв, з можливістю одночасно зберігати від 1 до 20 різних прийомів їжі та від 0 до 10000 найменувань страв і продуктів.
5. Можливість збереження меню з програми на 7 днів.

Мікропроцесорна система у складі програмно-апаратного комплексу повинна задовільняти наступні технічні вимоги:

1. Прийом та обробка сигналів переривання від зовнішніх пристроїв, кількістю, що дорівнює або більше 25 пристроїв;
2. Максимальна частота синхронізації більше або дорівнює 2 МГц;
3. Застосування напівпровідникового матричного індикатора розмірністю 8 рядків та 5 стовбців;
4. Частота розгортки зображення на індикаторі більше або дорівнює 100 Гц.

В ході розробки, для вирішення поставлених задач було використано об'єктно-орієнтоване програмування мовою C# [2], середовище розробки Microsoft Visual Studio [3] (рис. 1.2.1) та стандарт XML для зберігання налаштувань проєктів.

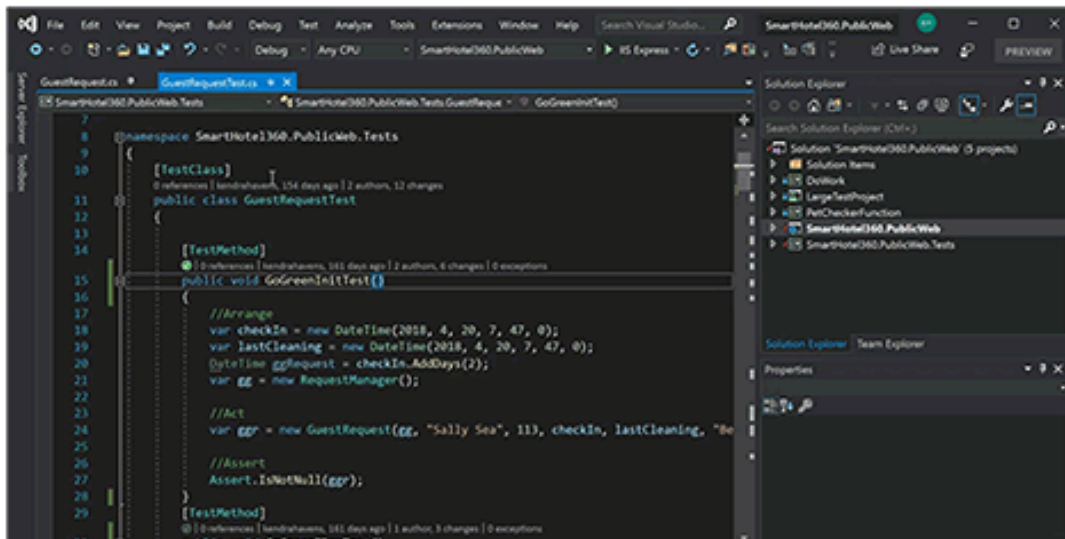


Рисунок 1.2.1 – Середовище програмування Microsoft Visual Studio 2019

Для написання бази даних до комплексу було використано Notepad++ (рис. 1.2.2). Створену базу було завантажено до Open Server (рис. 1.2.3), для створення локального веб-серверу з усією базою апаратного комплексу [4].

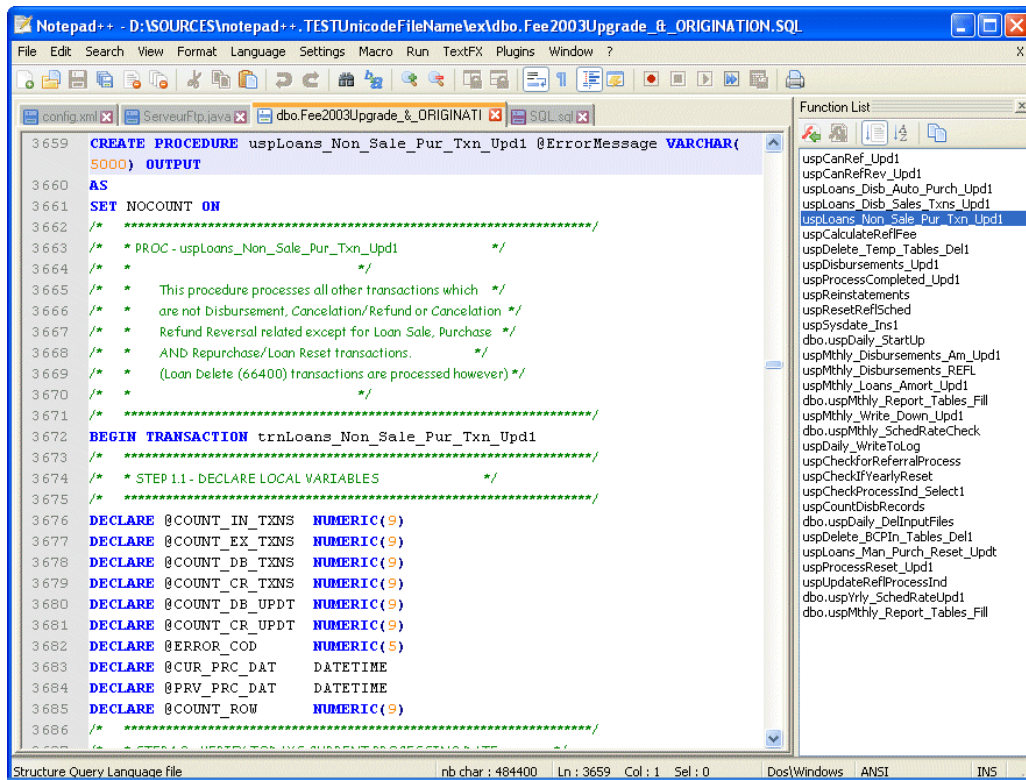


Рисунок 1.2.2 – Текстовий редактор Notepad++

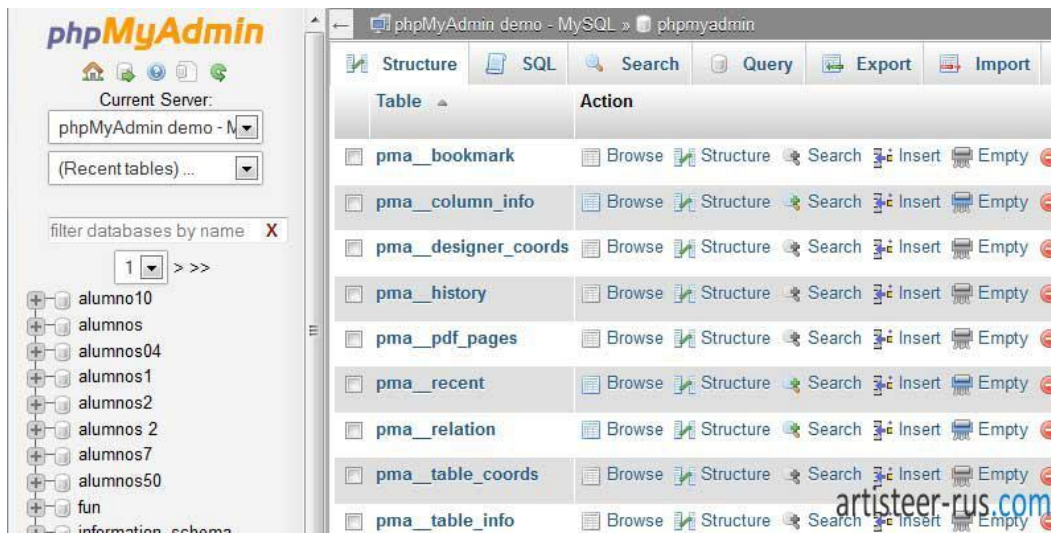


Рисунок 1.2.3 – Локальний веб сервер

Використання локального веб-серверу зумовлено його зручністю та мобільністю. Open Server можливо вільно завантажити та використовувати на безкоштовній основі, також даний сервер зручний у своєму використанні та сумісний з усіма операційними системами будь-якого персонального комп'ютера.

2 РОЗРОБЛЕННЯ АЛГОРИТМУ РОБОТИ ТА СТРУКТУРНОЇ СХЕМИ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ

2.1 Синтез алгоритму функціонування програмно-апаратного комплексу

Синтез програмно-апаратного комплексу для полегшення кодування було розбито на три основні частини:

1. Розробка вхідних даних.
2. Створення головних процесів.
3. Організація логіки програми за допомогою циклу.

Перша частина котра була розроблена – це вхідні дані, а саме база даних, котру у подальшому було підключено до Open Server, а також всі продукти лікарняного комплексу, для того аби додати їх у створений програмно-апаратний комплекс.

Наступним кроком створення даного комплексу була розробка головних вікон :

- Для додавання нових даних до бази даних.
- Для редагування вже існуючих даних.
- Для створення шаблону.
- Для виведення шаблону до друку.

Дані процеси доступні користувачу у будь-який момент роботи програмно-апаратного комплексу, завершення роботи додатку відбудеться лише тоді, коли буде натиснуто кнопку, що відповідає за припинення роботи комплексу.

Останнім етапом роботи було створення логіки програми за допомогою відповідних циклів. Основним циклом було обрано «do..while», тобто програмно-апаратний комплекс буде працювати до тих пір, поки не буде виконана задана умова, в даному випадку – поки користувач не завершить роботу з додатком.

Для реалізації переходу між різними вікнами комплексу було обрано цикл «if», тобто йде перевірка яке вікно обрав користувач. Наприклад, якщо було обрано редагування вже існуючих даних, то йде перевірка «Користувач обрав додавання нових даних?» → «Ні » → «Користувач обрав редагування вже існуючих даних?» → «Так» → Перехід до вікна, котре відповідає за

									Арк.
									12
Зм.	Лист	№ докум.	Підпис	Дата	ЕлІТ 6.171.00.10.415 ПЗ				

2.2 Синтез структурної схеми мікропроцесорної системи у складі програмно-апаратного комплексу

Побудова мікропроцесорної системи має здійснюватися на базі наступних принципів:

1. Функціональна завершеність застосовуваних в системі блоків;
2. шинна організація зв'язків;
3. регулярність структури мікропроцесорної системи.

Відповідно до зазначених принципів при практичній реалізації системи необхідно і доцільно орієнтуватися на трьохшину структуру і застосування програмно-керованих спеціалізованих контролерів.

Виходячи із завдання на проектування, мікропроцесорна система повинна включати в себе три підсистеми:

1. Підсистему управління роботою пристроїв мікропроцесорної системи;
2. підсистему обробки 25 переривань
3. підсистему виведення інформації на чотирьохрозрядний дискретний матричний індикатор.

Підсистема управління роботою пристроїв мікропроцесорної системи (далі підсистема управління) призначена для реалізації алгоритмів обробки інформації відповідно до набором функцій, які виконуються мікропроцесорної системою, і для управління роботою пристроїв системи відповідно до принципу програмного управління. Завданнями, які розв'язуються даною підсистемою, є:

1. Управління роботою всіх блоків мікропроцесорної системи;
2. програмна реалізація алгоритмів обробки даних і управління;
3. зберігання результатів обробки даних.

Для виконання завдань підсистемою управління в її склад має містити такі блоки:

1. Блок процесора;
2. блок пам'яті команд;
3. блок пам'яті даних.

Підсистема обробки переривань призначена для прийому, пріоритетної обробки і обслуговування запитів переривань. Завданнями, які розв'язуються даною підсистемою, є:

1. Прийом запиту переривань і формування сигналу переривання для підсистеми управління (блоку процесора);
2. заборона і разрешеніє прийому запитів переривань;
3. ідентифікація джерел запитів переривань;
4. передача управління підпрограм обслуговування запитів переривань;
5. збереження поточного стану блоку процесора при переході до підпрограм обслуговування запитів переривань і його відновлення після завершення підпрограми;
6. обробка пріоритетів запитів переривань;
7. сполучення засобів підсистеми обробки переривань з підсистемою управління.

Слід зазначити, що частину своїх завдань підсистема обробки переривань вирішує спільно з підсистемою управління. Для вирішення зазначених завдань підсистемою обробки переривань в її склад необхідно включити:

1. Блок прийому запитів переривань;
2. блок дозволу переривань;
3. блок обробки пріоритетів.

Підсистема виведення інформації на чотирьохрозрядний дискретний матричний індикатор (далі підсистема виведення) призначена для відображення результатів обробки даних. Завданнями, які розв'язуються даною підсистемою, є:

1. Прийом і зберігання даних, що відображаються на час циклу поновлення;
2. індикація результатів обробки даних.

Для вирішення зазначених завдань підсистемою виведення в її склад необхідно включити наступні блоки:

1. Блок прийому даних, що відображаються;
2. блок керування індикацією;
3. блок матричного індикатора.

					ЕЛІТ 6.171.00.10.415 ПЗ	Арк.
						15
Зм.	Лист	№ докум.	Підпис	Дата		

Структурна схема мікропроцесорної системи приведена на рисунку 2.2.2.

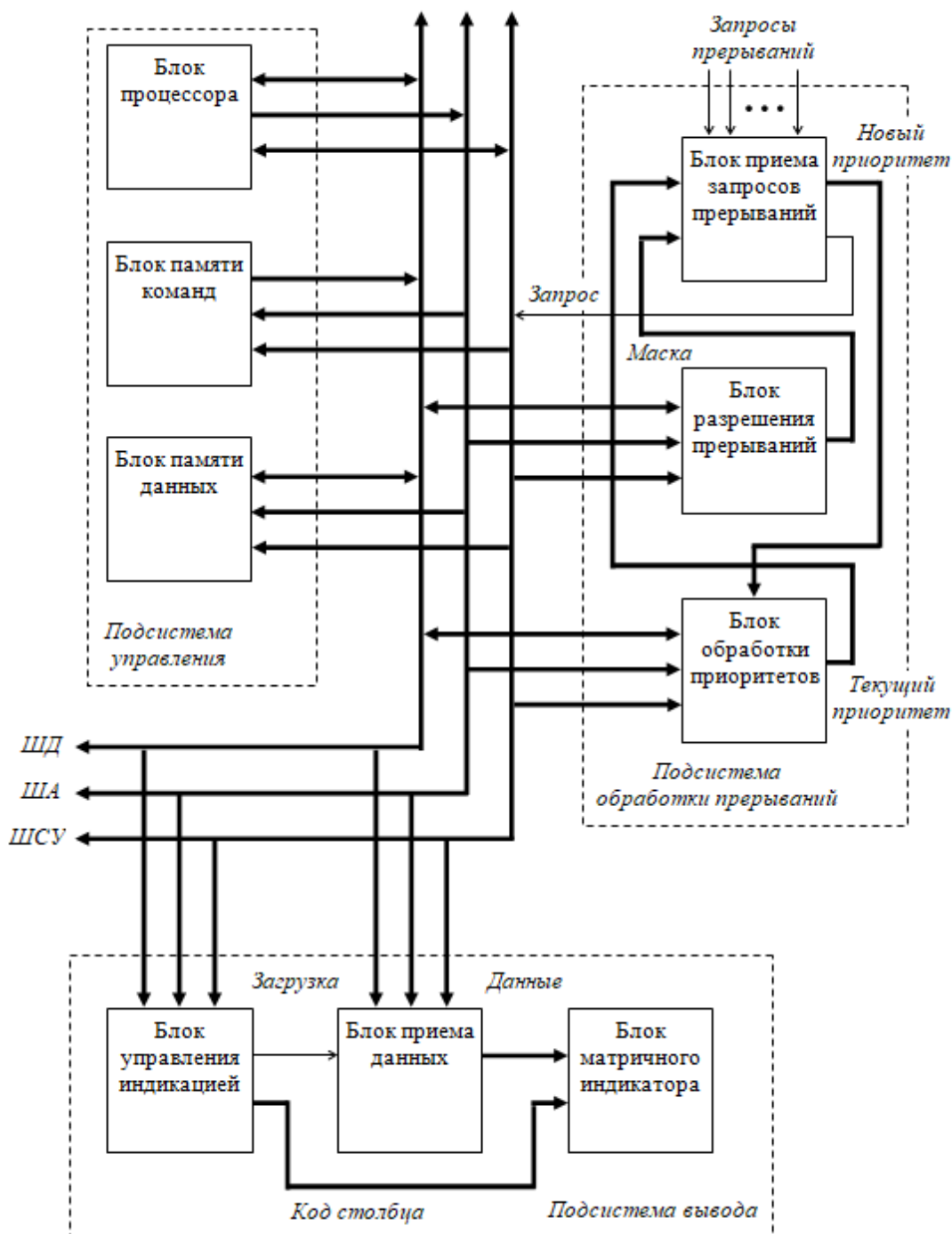


Рисунок 2.2.2 – Структурна схема мікропроцесорної системи

3 РОЗРОБКА ПРИНЦИПОВОЇ СХЕМИ МІКРОПРОЦЕСОРНОГО ПРИСТРОЮ КОНТРОЛЮ У СКЛАДІ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ

3.1 Вибір елементної бази мікропроцесорної системи

В якості основного елемента підсистеми управління пропонується використовувати керуючий мікроконтролер з сімейства PIC контролерів. В даний час існує безліч різних моделей цих контролерів, що відрізняються за своїми можливостями – кількістю і розрядністю портів введення / виводу, типом і обсягом вбудованої пам'яті і т.п.

В якості елементної бази для підсистеми обробки переривань доцільно застосувати існуючі контролери обробки переривань, наприклад БІС програмованого контролера переривань Intel 8259. Для управління цією мікросхемою необхідні 14 ліній введення-виведення (8 ліній – шина даних, 6 ліній – керуючі сигнали). Так як необхідно обслужити 25 переривань, а одна мікросхема контролера переривань здатна обслужити 8 переривань, то необхідно використовувати чотири мікросхеми Intel 8259. Отже, для управління чотирма контролерами переривань буде потрібно 17 ліній введення / виводу (8 – шина даних, 4 – вибір мікросхеми, 1 – вихід INT, 1 – введення адресного сигналу A0, 1 – введення керуючого сигналу INTA, 1 – введення керуючого сигналу RD, 1 – введення керуючого сигналу RD / WR).

Для технічної реалізації підсистеми виведення інформації на чотирьохрозрядний матричний дисплей необхідні 9 ліній введення / виводу (7 – ряди індикатора, 1 – введення керуючого сигналу RES, 1 – введення синхронізуємого сигналу CLK).

Таким чином, в сумі необхідно 26 ліній введення / виведення. З сімейства PIC вибираємо контролер PIC16F877, який має три восьмирозрядних, один шестирозрядний і один трьохрозрядний порти введення-виведення [5].

Схема керуючої підсистеми на базі мікроконтролера PIC16F877, що реалізує одночасно блоки процесора, пам'яті команд і пам'яті даних (див. Структуру системи на рисунку 2.2.2), показана на рисунку 3.1.1.

					ЕЛІТ 6.171.00.10.415 ПЗ	Арк.
						17
Зм.	Лист	№ докум.	Підпис	Дата		

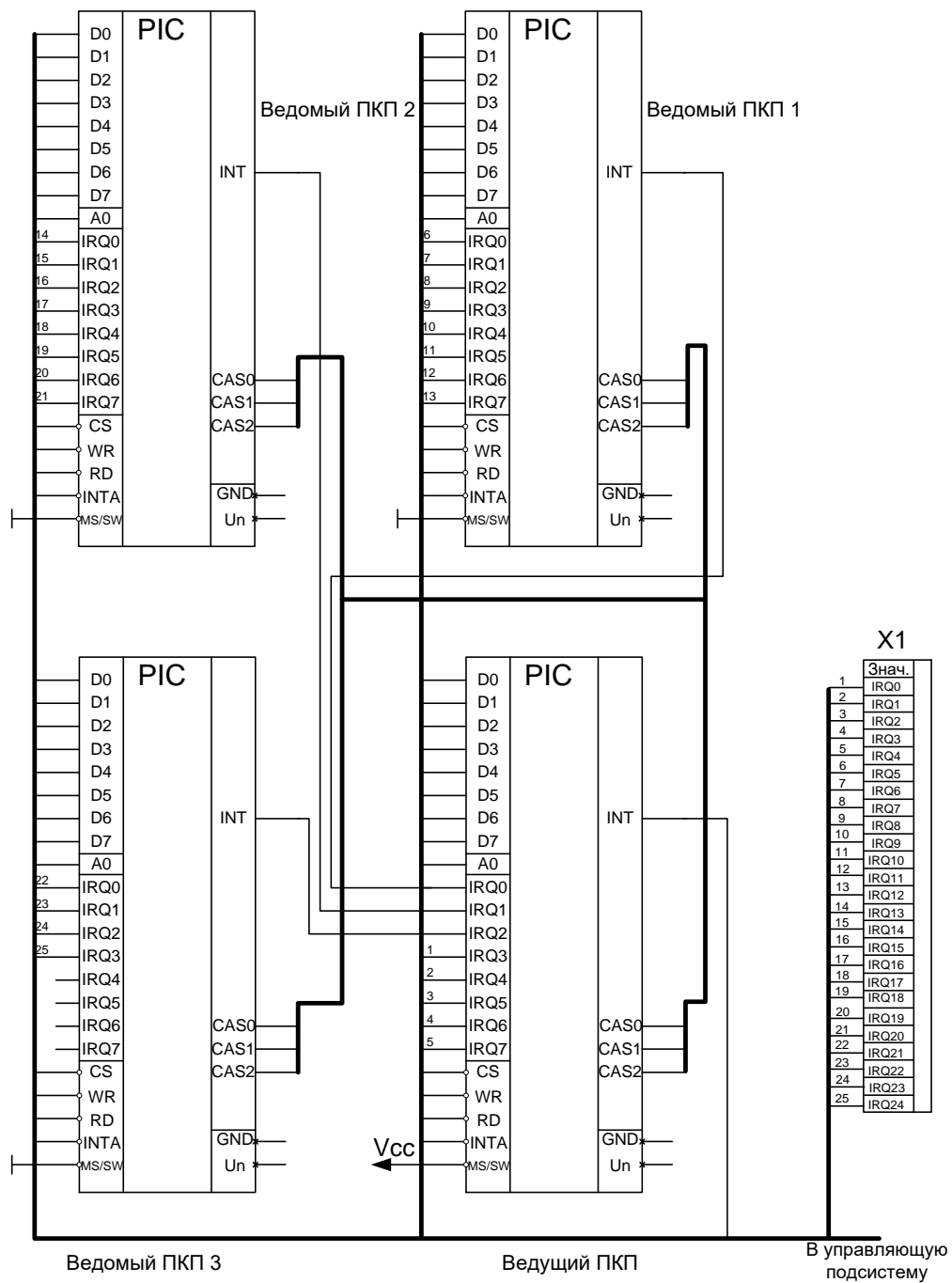


Рисунок 3.2.1 – Функціональна схема підсистеми обробки переривань

3.3 Розробка підсистеми виводу на матричний індикатор

Згідно з завданням, необхідно побудувати підсистему виводу даних на матричний дискретний індикатор з чотирьох розрядів. Розмірність одного розряду індикатора вибираємо 8*5 (8 рядків і 5 стовпців). Для побудови індикатора вибираємо світлодіодний матричний індикатор L835 / 4HDT,

кожен стовпець якого є лінією з чотирьох світлодіодних клітинок, а кожна клітинку в свою чергу складають чотири світлодіода [7].

Для управління таким індикатором застосовуємо схему динамічної індикації. Динамічна індикація характеризується тим, що імпульси розгортки, що подаються на стовпці (рядки) індикаторів, повинні мати шпаруватість $q \leq 1$. У розробляється підсистемі виведення застосовуємо постолбцовую розгортку. Частота кадру f_k для дискретних індикаторів повинна більше або дорівнює частоті $f_{кчм}$ критичного миготіння, має значення 50 Гц, тобто:

$$f_k = f_z / z \geq 50 \text{ Гц},$$

де z – число стовпців використовуваного матричного індикатора (в даній підсистемі 20);

f_z – частота імпульсів розгортки.

Приймаємо $f_k = 50$ Гц. Використовуючи останню формулу, визначаємо частоту імпульсів розгортки:

$$f_z \geq f_k \cdot z = 50 \cdot 20 = 100 \text{ Гц}.$$

Таким чином, при розробці програмного забезпечення мікропроцесорної системи необхідно передбачити програмну реалізацію формування імпульсів розгортки з частотою $f_z = 100$ Гц (сигнал CLK через висновок PB4 мікроконтролера PIC16F877 на рисунку 3.1.1).

Функціональна схема підсистемі виведення на чотирьохрозрядний матричний індикатор показана на рисунку 3.3.1. Розподілом імпульсів розгортки на стовпці індикатора займаються лічильник DD1, мультиплексор DD2 і дешифратори DD5-DD6. З молодшого розряду лічильника DD1.1 на вхід-Е DD3 і вхід Е DD4 буферних підсилювачів подаються імпульси дозволу виведення кодів рядків. Елементи DD3-DD4 забезпечують високу навантажувальну здатність для індикаторної рядки (її половини). Імпульси дозволу На сусідні горизонтальні світлодіоди в стовпці подаються однакові рівні сигналів. На малюнку приведено підключення тільки чотирьох стовпців світлодіодної матриці, інші 16 стовпців підключаються аналогічно (до виходів дешифраторів, зазначеними цифрами 3-10).

3.4 Розробка програмного забезпечення для мікропроцесорної системи

Пристрій працює наступним чином:

1. Після включення живлення (або скидання) мікроконтролер виконує початкову настройку своїх портів введення / виводу, підключає дільник до таймера TMR0 і встановлює його коефіцієнт, що дорівнює 1/64 (при такому значенні дільника переповнення таймера відбувається приблизно 125 раз в секунду), а також робить налаштування всіх БІС Intel 8259 і потім приступає до виконання основної програми.

2. При переповненні таймера TMR0 відбувається переривання (з установкою прапора T01F). Оброблювач переривань перевіряє установку прапора T01F і при встановленому прапорі викликає підпрограму виведення зображення на матричних індикаторах.

При виникненні переривання від зовнішнього пристрою підсистема обробки переривань визначає номер пристрою, що викликав переривання, і викликає відповідну підпрограму обробки переривання.

Лістинг керуючої програми апаратної частини комплексу наведено в додатку А.

Перелік використаних елементів наведено в додатку Б.

					ЕЛІТ 6.171.00.10.415 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		22

4 РОЗРОБКА ЗАГАЛЬНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ

4.1 Проектування загального програмного забезпечення

Під час виконання дипломного проєкту було використано методологію IDEF0, тобто методологію функціонального моделювання [8]. За допомогою даної методології є можливість показати не тільки ресурси, але й інструменти, котрі були використані для реалізації процесу.

На рисунках 4.1.1 – 4.1.2 наведено контекстну діаграму та діаграму декомпозиції створення програмно-апаратного комплексу документування даних лікарняних закладів. IDEF0-діаграма описує процес створення комплексу у вигляді блоків, що можна декомпозувати.

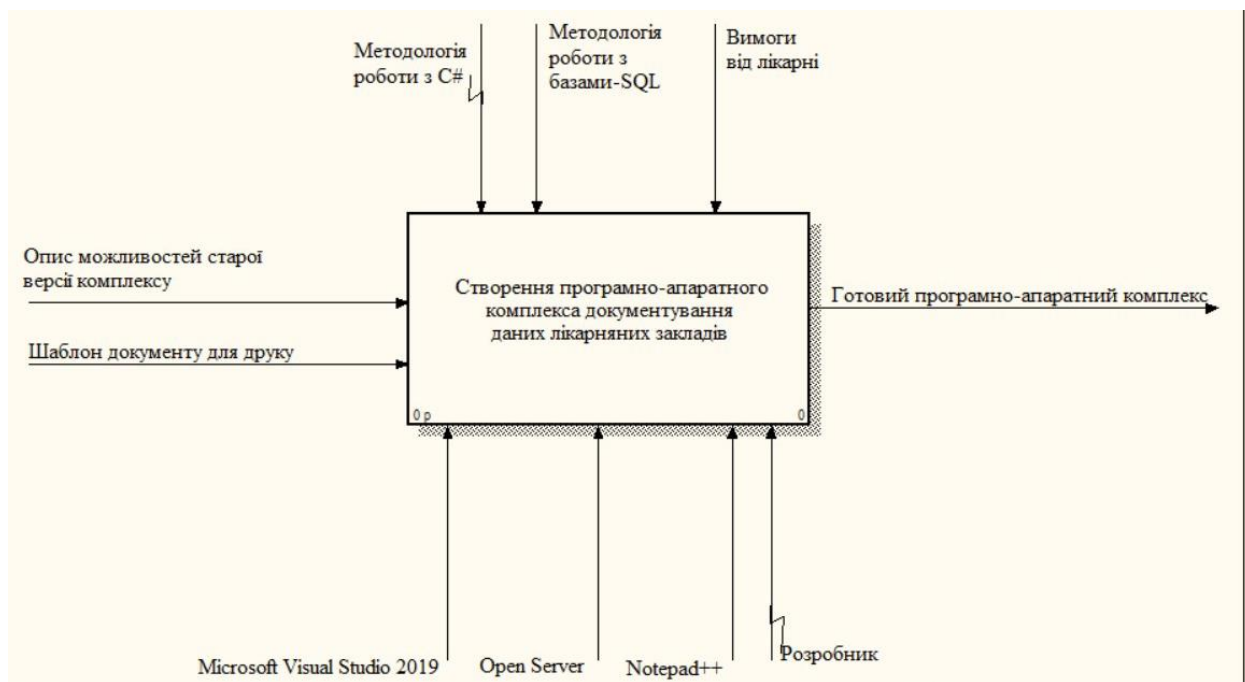


Рисунок 4.1.1 – Контекстна діаграма

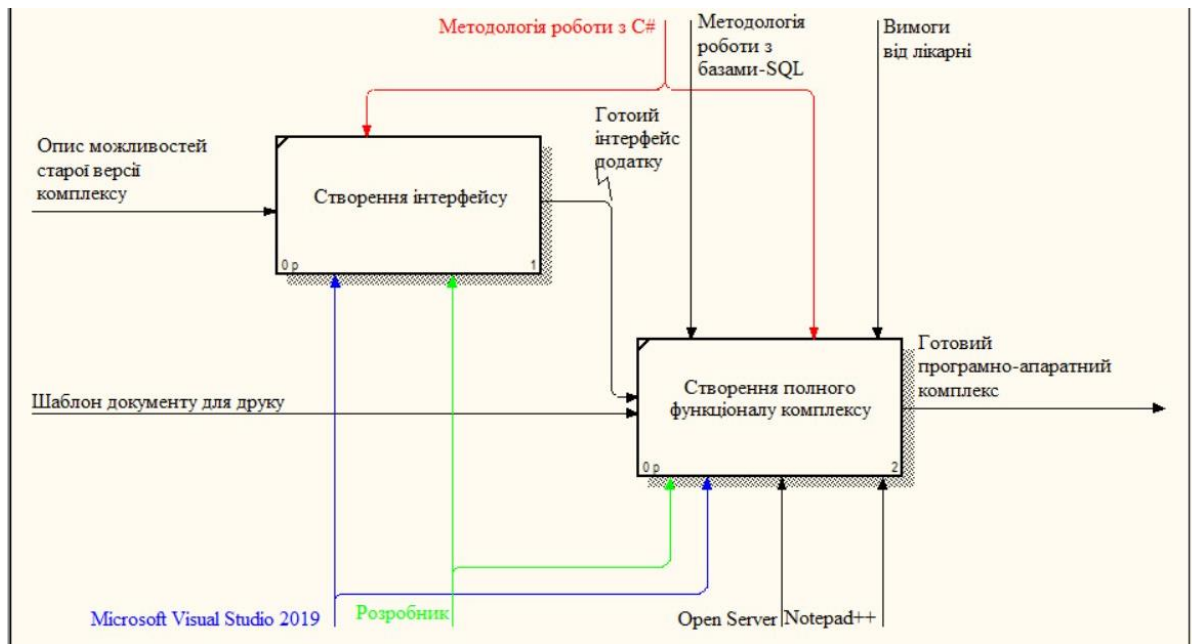


Рисунок 4.1.2 – Декомпозиції діаграми IDEF0

Далі було проведена декомпозиція одного із основних блоків діаграми, а саме «Створення повного функціоналу комплексу». На рисунку 4.1.3 наведена діаграма «Створення повного функціоналу комплексу».

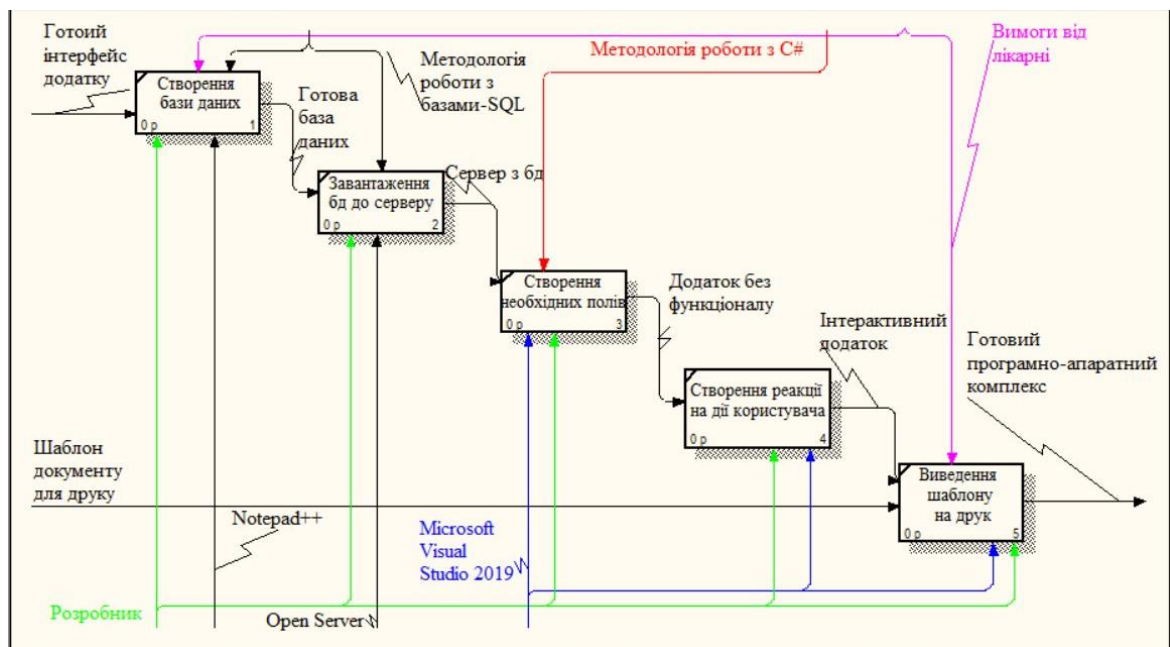


Рисунок 4.1.3 – Декомпозиція блоку «Створення повного функціоналу комплексу»

4.2 Пояснення до програми

Під час першого запуску розробленої програми необхідно підключити базу даних до веб сервера, це треба зробити лише один раз. Для цього необхідно запустити веб-сервер «Open Server», потім в системному треї потрібно натиснути на ярлик «Open Server», відкриється діалогове вікно зображене на рисунку 4.2.1. Наступним кроком в відкритому вікні натискаємо клавішу «Запустить», для старту веб-серверу.

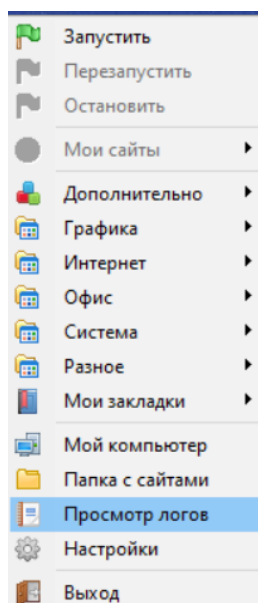


Рисунок 4.2.1 – Контекстне меню запуску Open Server

Далі необхідно увійти до додатку «PhpMyAdmin», через контекстне меню, як це зробити зображено на рисунку 4.2.2 [9].

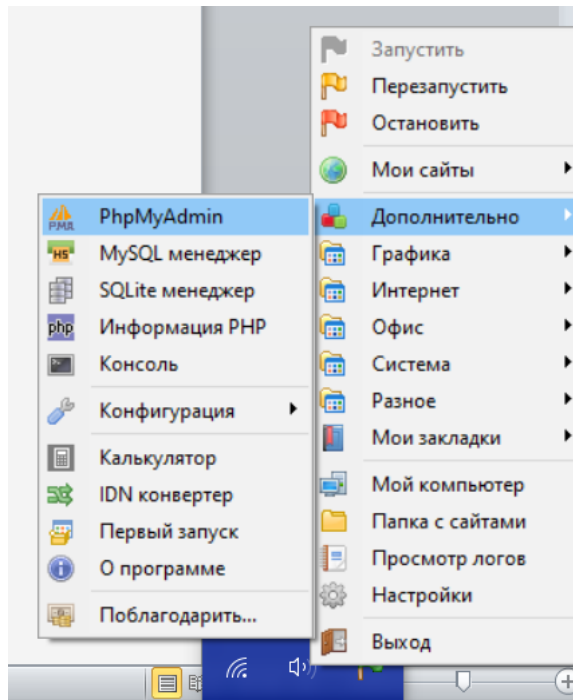


Рисунок 4.2.2 – Контекстне меню запуску додатку «PhpMyAdmin»

Після цього відкриється сторінка авторизації додатку «PhpMyAdmin», вводимо логін «root», пароль залишаємо порожнім (рис 4.2.3) та натискаємо клавішу «Вперед».

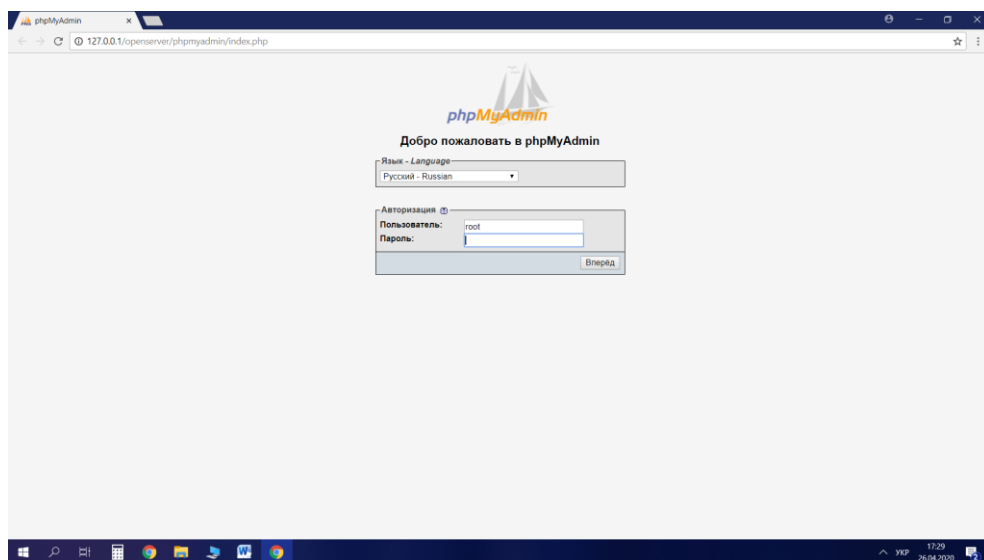


Рисунок 4.2.3 – Веб сторінка авторизації в «PhpMyAdmin»

Відкриється головна сторінка «PhpMyAdmin» (рис 4.2.4), в верхній частині знаходимо клавішу «Імпорт», натискаємо та переходимо до вікна імпорту бази даних (рис 4.2.5).

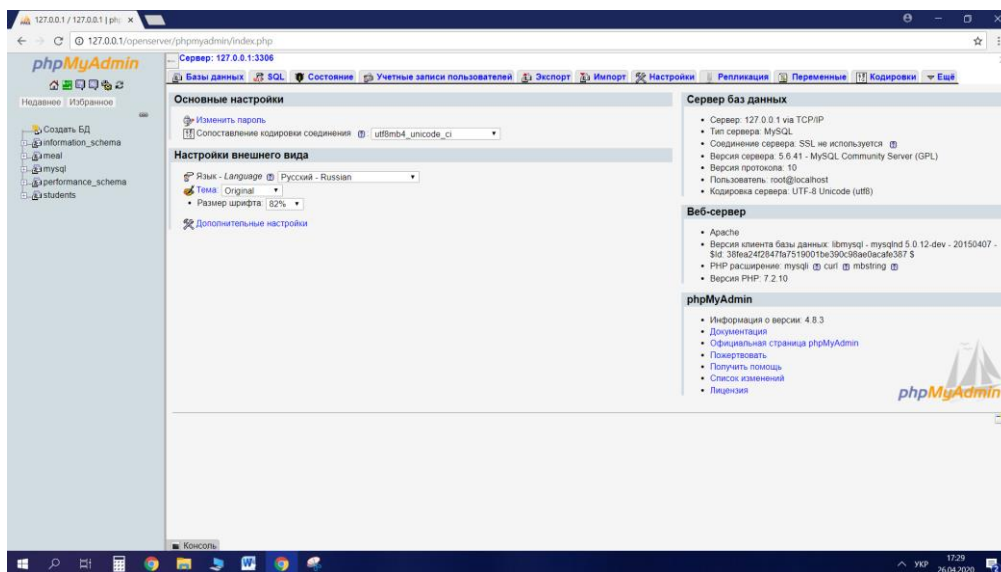


Рисунок 4.2.4 – Головна сторінка «PhpMyAdmin»

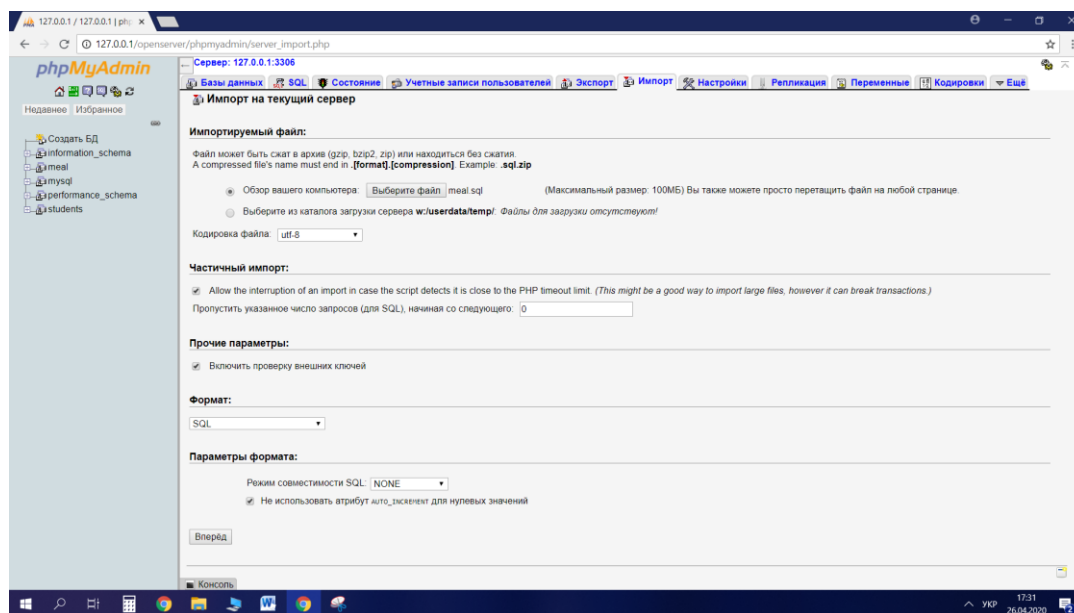


Рисунок 4.2.5 – Сторінка імпорту бази даних в «PhpMyAdmin»

Необхідно натиснути «Выберете файл» та у відкритому вікні (рис 4.2.6) вибрати базу даних з назвою «meal.sql». Після чого натискаємо «Открыть». База даних успішно імпортована.

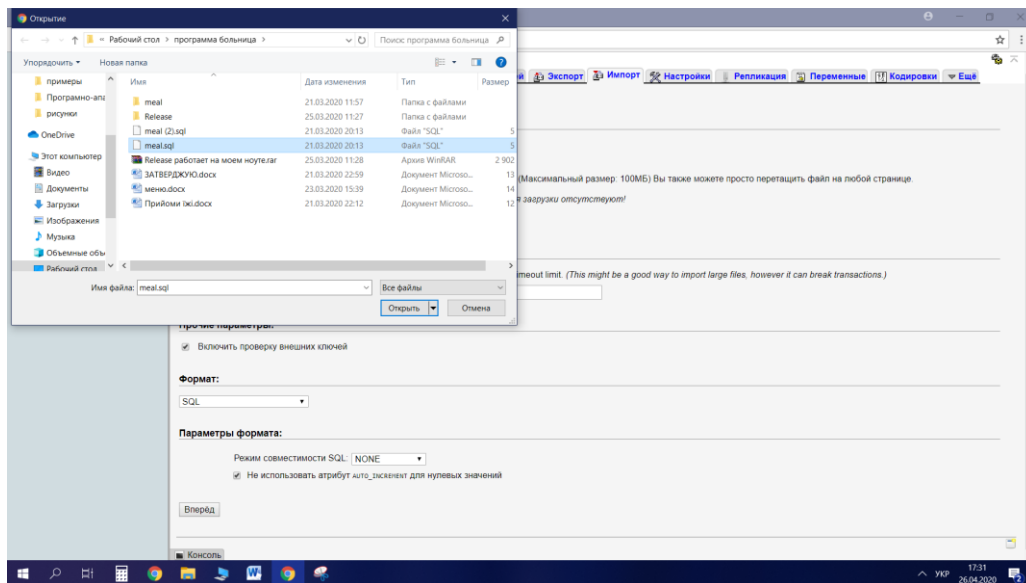


Рисунок 4.2.6 – Вікно вибору файлу, який буде імпортовано

Переходимо до запуску самої програми, для цього відкриваємо папку в якій вона зберігається та запускаємо файл «meal.exe» (рис 4.2.7), після чого відкриється головне вікно програми, з назвою «Прийоми їжі» (рис 4.2.8)

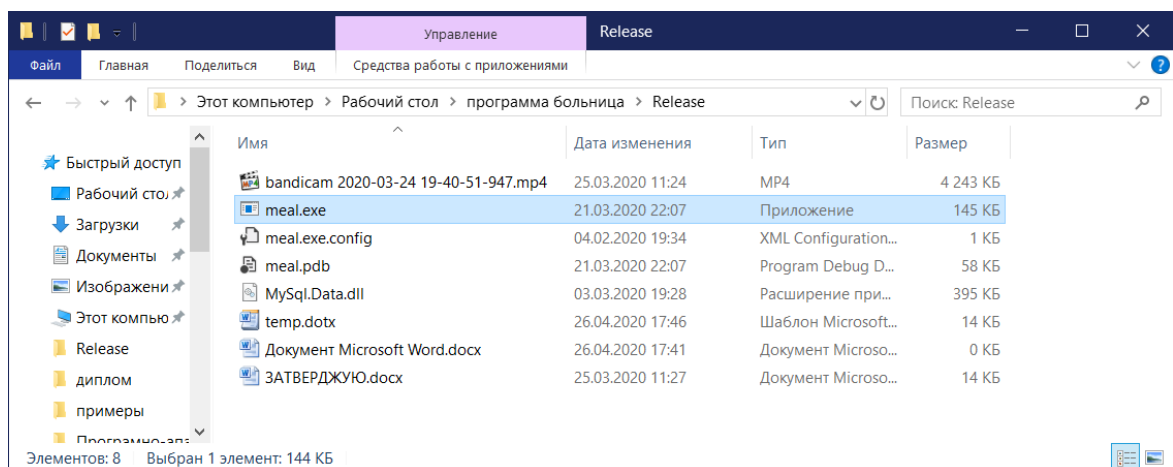


Рисунок 4.2.7 – Запуск розробленої програми

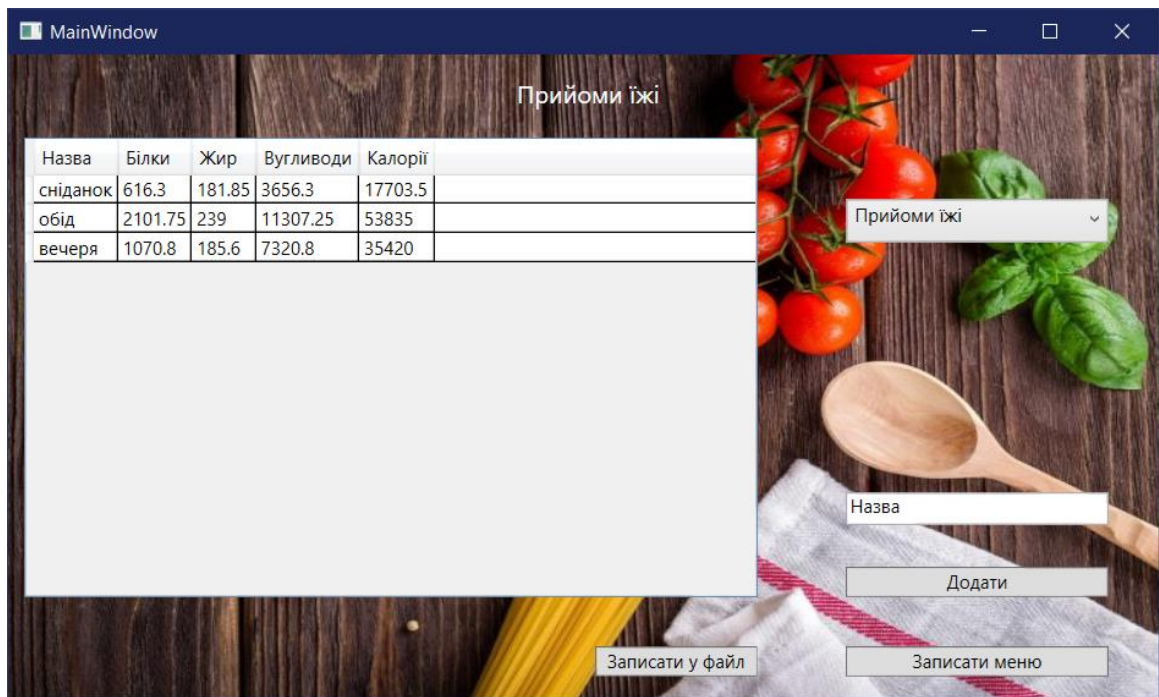


Рисунок 4.2.8 – Головне вікно програми під назвою «Прийоми їжі»

Це вікно (рис 4.2.8) складається з клавiшi «прийоми їжі» в правiй частинi вiкна, при натисканнi на неї вiдкривається розкривне меню (рис 4.2.9) з перелiком основних вiкон програми, а саме: «Прийоми їжі», «Страви», «Продукти». В вiкнi «Прийоми їжі» вiдображуються всi доданi прийоми їжі, їх загально калорiйнiсть та склад бiлків, жирiв та вуглеводiв (БЖУ). Для додання нового прийому їжі необхідно замiсть слова «Назва» ввести бажану назву прийому їжі та натиснути клавiшу «Додати», пiсля чого прийом їжі буде додано та збережено.

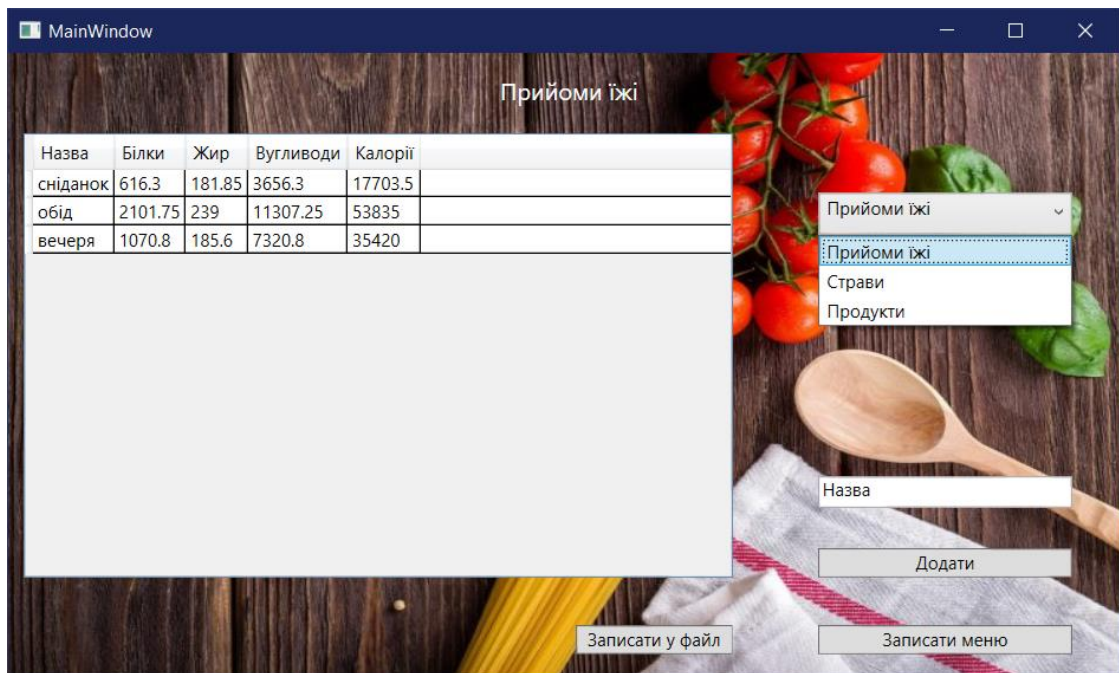


Рисунок 4.2.9 – Головне вікно після натискання клавіші «прийоми їжі»

При відкритті вікна «Страви» (рис 4.2.10) бачимо перелік всіх створених страв, їх вагу, склад білків, жирів, вуглеводів та калорійність. Додавання нової страви аналогічне додаванню нового прийому їжі.

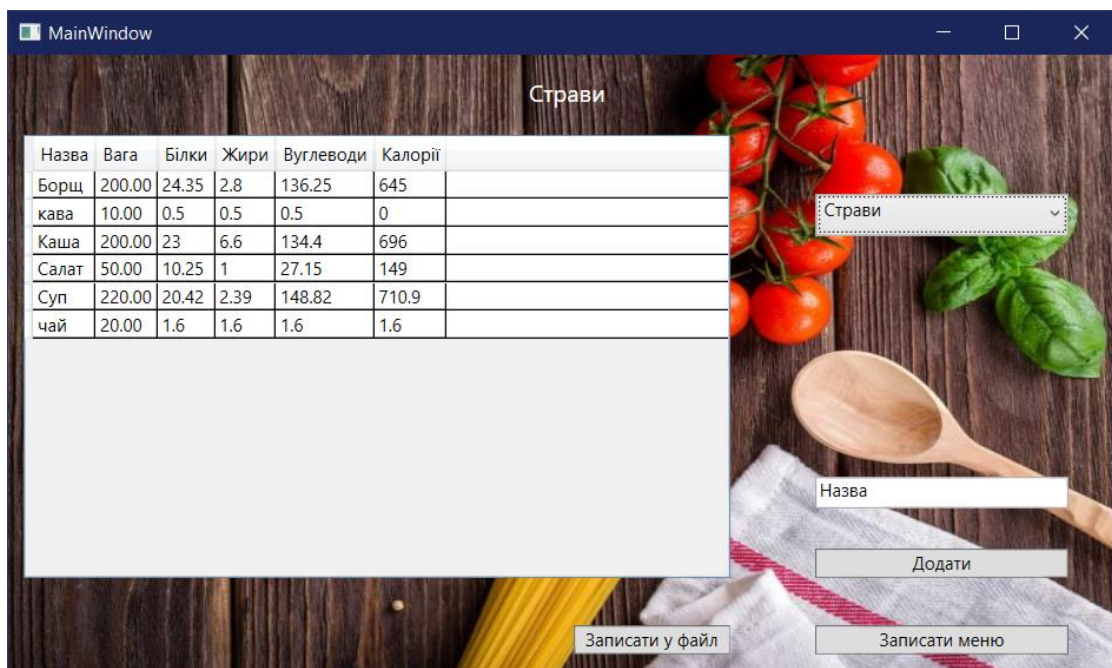


Рисунок 4.2.10 – Вигляд вікна «Страви»



Рисунок 4.2.11 – Вигляд вікна «Продукти»

На рисунку 4.2.11 зображено вікно «Продукти», в якому зберігається перелік всіх доданих продуктів, з їх харчовою цінністю та складом БЖУ. Додавання нового продукту відбувається внизу списку, в порожньому полі, яке виділено синім на рисунку 4.2.11, вводиться назва продукту, його БЖУ та калорії. Після натискається клавіша «Додати». Для збереження змін натискаємо клавішу «Зберегти зміни» та відповідно для скасування «Скасувати зміни».

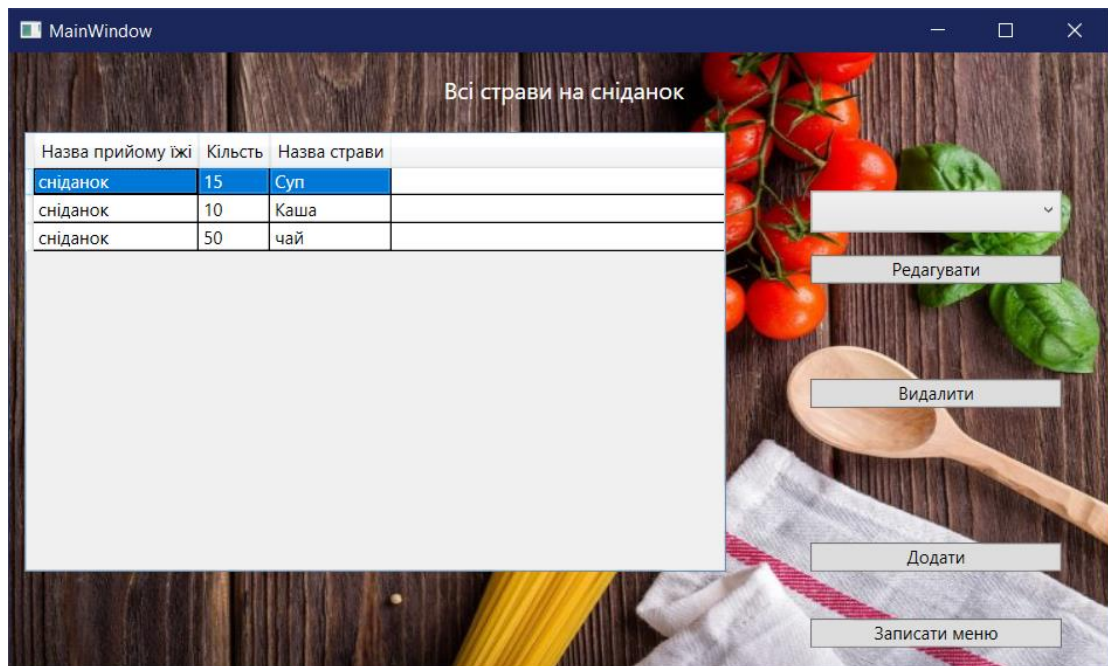


Рисунок 4.2.12 – Вікно перегляду страв, доданих до прийому їжі «Сніданок»

Якщо в вікні «Прийоми їжі» вибрати будь який прийом їжі та натиснути клавішу «Переглянути», то буде відкрито вікно «Всі страви на ...» (рис 4.2.12). В якому видно всі страви які було додано до прийому їжі «сніданок» та їх кількість. Страви можна додавати, видаляти та редагувати, змінюючи їх кількість, за що відповідають клавіші «Додати», «Видалити», «Редагувати».

Якщо в вікні «Страви» натиснути на одну зі страв то з'являться клавіші «Переглянути» та «Видалити» (рис 4.2.13).

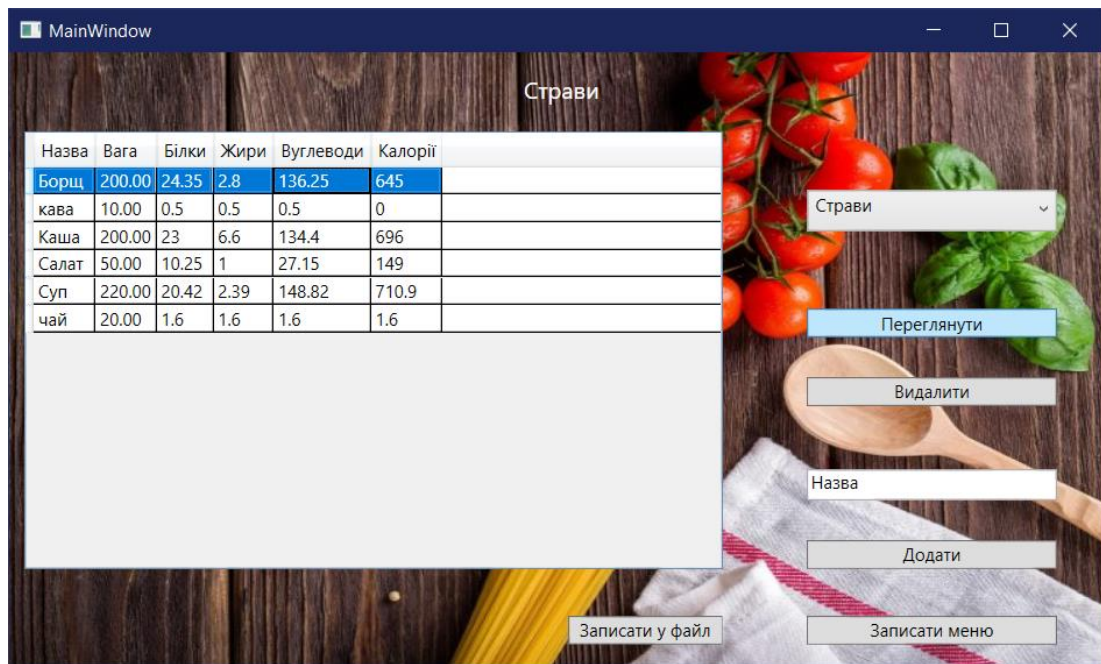


Рисунок 4.2.13 – Вікно «Страви» під час вибор однієї зі страв

Якщо в вікні «Страви» натиснути на одну зі страв то з'являться клавiші «Переглянути» та «Видалити» (рис 4.2.13). При натисканні клавiші «Переглянути» відкривається вікно переліку продуктів доданих до страви (рис 4.2.14), де зображено яку страву відкрито, назву продукту який входить в склад страви та його вагу. При виборі одного з продуктів з'являється клавiші «Редагувати» та «Видалити», аналогічно як і у вікні перегляду страв, доданих до прийому їжі.

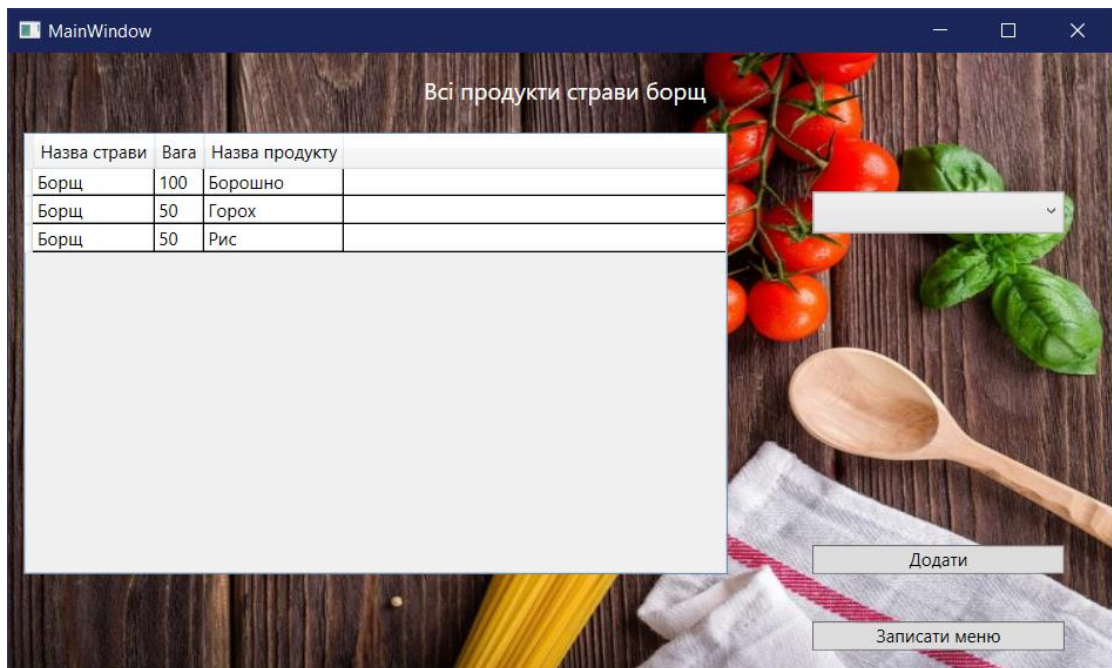


Рисунок 4.2.14 – Вікно переліку продуктів доданих до страви «Борщ»

При натисканні клавіші «Записати меню» відкріється діалогове вікно «Сохранение документа» (рис 4.2.15), яке звертаючись до файлу шаблону temp.dotx, дає змогу вибрати куди та з якою назвою зберігти розроблене меню. За умовчужанням назва файлу буде «ЗАТВЕРДЖУЮ», при необхідності її можна змінити.

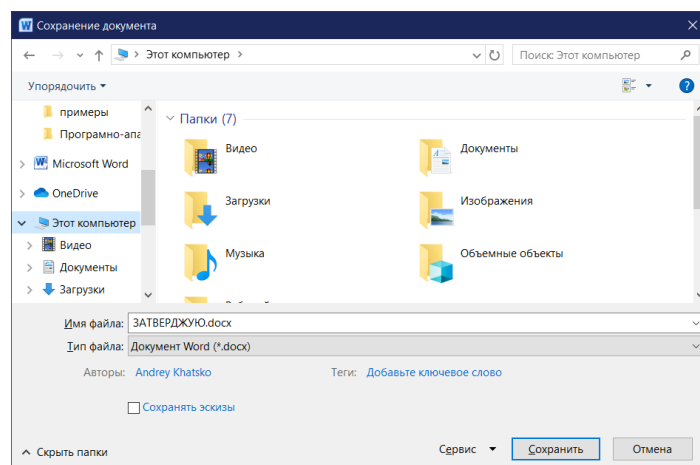


Рисунок 4.2.15 – Діалогове вікно «Сохранение документа» для збереження ГОТОВОГО МЕНЮ

Збережений файл (рис 4.2.16) містить в собі таблицю з переліком всіх прийомів їжі, перелік страв та їх кількість, з яких складаються ці прийоми,

перелік продуктів з яких складається кожна страва та вага кожного продукту в кожній страві в грамах. Зверху таблиці поля для підпису головного лікаря лікарні та дати коли це меню дійсне. Знизу таблиці поля для підписів дієтсестри, кухаря, дієтолога, шеф-кухара та бухгалтера.

Меню	Кількість	Борошно	Горох	Рис	Хліб житній	Макаронні вироби	Пашоно	чай чорний	цукор	Б	Ж	В	Калорії	Витд	Підпис
сніданок															
Каша	10						200			23	6,6	134,4	696	200,00	
Суп	15			50	20	150				20,42	2,39	148,82	710,9	220,00	
чай	50							10	10	1,6	1,6	1,6	1,6	20,00	
Всього				750	300	2250	2000	500	500	616,3	181,85	3656,3	17703,5		
обід															
Борщ	80	100	50	50						24,35	2,8	136,25	645	200,00	
кава	60									0	0	0	0	0,00	
Салат	15		50							10,25	1	27,15	149	50,00	
Всього		8000	4750	4000						2101,75	239	11307,25	53835		
вечера															
Каша	10						200			23	6,6	134,4	696	200,00	
Суп	40			50	20	150				20,42	2,39	148,82	710,9	220,00	
чай	15							10	10	1,6	1,6	1,6	1,6	20,00	
Всього				2000	800	6000	2000	150	150	1070,8	185,6	7320,8	35420		

Рисунок 4.2.16 – Відкритий файл готового меню під назвою «ЗАТВЕРДЖУЮ»

4.3 Розробка програмної частини програмно-апаратного комплексу

Для того щоб створити даний комплекс, перш за все, було написано базу даних на мові SQL, яка містила у собі всю початкову необхідну інформацію. Фрагмент коду створення бази даних зображено на рисунку 4.3.1, повний лістинг програми міститься у додатку В.

```

meal (2).sql x
1  -- phpMyAdmin SQL Dump
2  -- version 4.8.3
3  -- https://www.phpmyadmin.net/
4  --
5  -- Хост: 127.0.0.1:3306
6  -- Время создания: Фев 14 2020 г., 00:50
7  -- Версия сервера: 8.0.12
8  -- Версия PHP: 5.5.38
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 SET AUTOCOMMIT = 0;
12 START TRANSACTION;
13 SET time_zone = "+00:00";
14
15
16 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
17 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
18 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
19 /*!40101 SET NAMES utf8mb4 */;
20
21 --
22 -- База данных: `meal`
23 --
24
25 -----
26
27 --
28 -- Структура таблицы `dish`
29 --
30
31 CREATE TABLE `dish` (
32   `dish_id` int(11) NOT NULL,
33   `name` varchar(25) NOT NULL
34 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
35

```

Рисунок 4.3.1 – Фрагмент коду бази даних

Після написання бази даних, файл було завантажено до локального серверу (рис. 4.3.2), аби мати можливість зчитувати та записувати інформацію до комплексу.

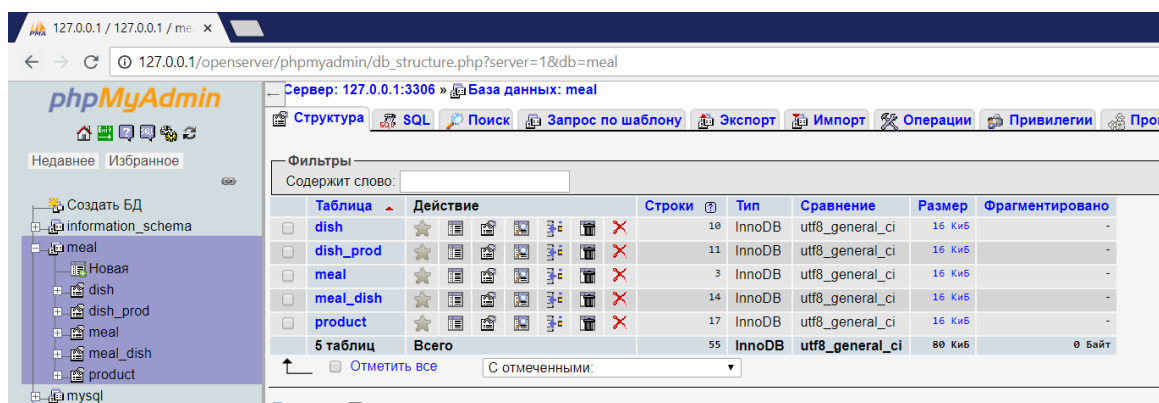


Рисунок 4.3.2 – Завантажена база даних до серверу

Частина коду для підключення до бази даних наведена на рисунку 4.3.3. Файл add.html.cs відповідає за роботу з базою даних, а саме додавання нової інформації до неї.

Головне меню (початкове вікно комплексу) описано у файлі MainWindow.xaml.cs, фрагмент коду оновлення таблиць у початковій таблиці наведено на рисунку 4.3.3.

```
void UpdateDataTable(DataTable d, String sqlq){
    sqlConnect.Open();
    MySqlCommand createCommand = new MySqlCommand(sqlq, sqlConnect);
    dataAdapter = new MySqlDataAdapter(createCommand);
    dataAdapter.Fill(d);
    sqlConnect.Close();
}
```

Ссылка: 3

```
void UpdateTable()
{
    try {
        dt = new DataTable();
        UpdateDataTable(dt, Sqlquery());
        dataGrid.ItemsSource = dt.DefaultView;
        dataGrid.Columns[0].Visibility = Visibility.Hidden;
        dt.Columns[0].AutoIncrement = true;
    }
    catch (Exception)
    {
        MessageBox.Show($"Помилка підключення до бази даних");
        Close();
    }
}
```

Рисунок 4.3.3 – Фрагмент коду головної сторінки

Для зміни вже існуючої інформації у базі даних використовується код Edit.xaml.cs, фрагмент лістингу програми для оновлення інформації наведено на рисунку 4.3.4.

```

private void Button_Click_1(object sender, RoutedEventArgs e)
{
    if (meal.MainWindow.isMeal) {
        SqlNonRequestQuery($"UPDATE meal_dish SET count={count.Text} WHERE md_id = {meal.MainWindow.editID};");
    }
    else {
        SqlNonRequestQuery($"UPDATE dish_prod SET weight={count.Text} WHERE dp_id = {meal.MainWindow.editID};");
    }
    Close();
}

```

Рисунок 4.3.4 – Фрагмент коду оновлення інформації

Для формування зовнішнього вигляду вікон програмного комплексу ієкристовувалися файли з розширенням .xaml. Головне вікно має назву MainWindow.xaml, вікно для редагування – Edit.xaml та для додавання нової інформації – add.xaml. Нижче наведено лістинг усіх головних вікон проєкту.

add.xaml

```

<Window x:Name="addWindow" x:Class="meal.add"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
    xmlns:local="clr-namespace:meal"
    mc:Ignorable="d"
    Title="add"                Height="199.082"                Width="422.68"
    Activated="firstActivated">
    <Grid>
        <ListBox                HorizontalAlignment="Left"                Height="100"
    Margin="486,200,0,0" VerticalAlignment="Top" Width="100"/>
        <TextBox x:Name="cont" HorizontalAlignment="Left" Height="23"
    Margin="241,76,0,0" TextWrapping="Wrap" VerticalAlignment="Top"
    Width="120"/>

```

```

        <Label                x:Name="boxlabel"                Content="Lable"
HorizontalAlignment="Left"  Margin="241,45,0,0"  VerticalAlignment="Top"
RenderTransformOrigin="-3.019,0.328" Width="120"/>
        <ComboBox  x:Name="addComboBox"  HorizontalAlignment="Left"
Margin="66,76,0,0" VerticalAlignment="Top" Width="120"/>
        <Button                x:Name="addBtn"                Content="Додати"
HorizontalAlignment="Left"  Margin="273,126,0,0"  VerticalAlignment="Top"
Width="88" Click="AddBtn_Click"/>
        <DataGrid  x:Name="dataGrid"  HorizontalAlignment="Left"
Height="36" VerticalAlignment="Top" Width="100"/>
        <Label                x:Name="comboLable"                Content="Label"
HorizontalAlignment="Left"  Margin="66,45,0,0"  VerticalAlignment="Top"
Width="120"/>
        <Label                x:Name="meinLabel"                Content="Label"
HorizontalContentAlignment="Center"                HorizontalAlignment="Left"
Margin="10,10,0,0" VerticalAlignment="Top" Width="395"/>
        <Button                x:Name="cancel"                Content="Скасувати"
HorizontalAlignment="Left"  Margin="150,126,0,0"  VerticalAlignment="Top"
Width="75" Click="Cancel_Click"/>
    </Grid>
</Window>

```

Edit.xaml

```
<Window x:Class="meal.Edit"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
  xmlns:local="clr-namespace:meal"
  mc:Ignorable="d"
  Title="Edit"           Height="199.485"           Width="394.701"
  Activated="firstActivated">
  <Grid>
    <TextBox x:Name="name" HorizontalAlignment="Left" Height="23"
  Margin="33,72,0,0"           TextWrapping="Wrap"           Text="TextBox"
  VerticalAlignment="Top" Width="120" IsEnabled="False"/>
    <TextBox x:Name="count" HorizontalAlignment="Left" Height="23"
  Margin="215,72,0,0"           TextWrapping="Wrap"           VerticalAlignment="Top"
  Width="120"/>
    <Button Content="ЗМІНИТИ" HorizontalAlignment="Left"
  Margin="260,126,0,0"           VerticalAlignment="Top"           Width="75"
  Click="Button_Click_1"/>
    <Button Content="Скасувати" HorizontalAlignment="Left"
  Margin="137,126,0,0"           VerticalAlignment="Top"           Width="75"
  Click="Button_Click"/>
    <Label x:Name="mainLabel" Content="Label"
  HorizontalContentAlignment="Center" HorizontalAlignment="Left"
  Margin="10,10,0,0" VerticalAlignment="Top" Width="367"/>
    <Label x:Name="nameLabel" Content="Label"
  HorizontalAlignment="Left" Margin="33,41,0,0" VerticalAlignment="Top"
  Width="120"/>
    <Label x:Name="countLabel" Content="Label"
  HorizontalAlignment="Left" Margin="215,41,0,0" VerticalAlignment="Top"
  Width="120"/>
```

					ЕЛІТ 6.171.00.10.415 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		40

</Grid>
</Window>

					ЕЛІТ 6.171.00.10.415 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		41

MainWindow.xaml

```
<Window x:Name="Wind" x:Class="meal.MainWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
  xmlns:local="clr-namespace:meal"
  mc:Ignorable="d"
  Title="MainWindow" Height="459" Width="767" Activated="Activ_
SizeChanged="Resize_wind">
  <Grid>
    <Grid.Background>
      <ImageBrush ImageSource="meals.jpg"/>
    </Grid.Background>
    <DataGrid HorizontalAlignment="Left" Height="301"
Margin="10,54,0,0" VerticalAlignment="Top" Width="480" x:Name="dataGrid"
CurrentCellChanged="CurrentCell" LostFocus="dataGridLostFocus"/>
    <ComboBox x:Name="selectList" HorizontalAlignment="Left"
Margin="548,94,0,0" VerticalAlignment="Top" Width="172" Height="29"
DropDownClosed="selectListClosed" SelectedIndex="0">
      <ComboBoxItem Content="Прийоми їжі"/>
      <ComboBoxItem Content="Страви"/>
      <ComboBoxItem Content="Продукти"/>
    </ComboBox>
    <Button x:Name="Commit" Content="Зберегти зміни"
HorizontalAlignment="Left" Margin="204,387,0,0" VerticalAlignment="Top"
Width="92" Click="Commit_Click"/>
    <Button x:Name="Save" Content="Записати у файл"
HorizontalAlignment="Left" Margin="384,387,0,0" VerticalAlignment="Top"
Width="106" Click="saveClick"/>
```

						ЕЛІТ 6.171.00.10.415 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата			42

<DataGrid x:Name="temp" HorizontalAlignment="Left" Height="100" Margin="28,360,0,-32" VerticalAlignment="Top" Width="100"/>

<DataGrid x:Name="prod" HorizontalAlignment="Left" Height="100" Margin="28,355,0,-27" VerticalAlignment="Top" Width="100"/>

</Grid>

</Window>

					ЕЛІТ 6.171.00.10.415 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		44

ВИСНОВКИ

Під час виконання даної кваліфікаційної роботи бакалавра було проведено повний огляд та аналіз програмно-апаратних аналогів. Сформована чітка постановка завдань, котрі необхідно було виконати у ході розробки проєкту.

Окрім опису аналогів було обрано програмні продукти за допомогою яких розроблявся даний програмно-апаратний комплекс. До переліку даних інструментів реалізації входять такі додатки як – Microsoft Visual Studio 2019, Notepad++, Open Server.

Під час розроблення алгоритму та структурної схеми програмно-апаратного комплексу було виконано декілька схем. Перш за все було сформовано схему алгоритму функціонування програмно-апаратного комплексу, котра поетапно показує, як саме налаштована логіка даного проєкту.

Також була створена структурна схема мікропроцесорної системи та розроблена принципова схема мікропроцесорного пристрою контролю у складі програмно-апаратного комплексу, з такими технічними параметрами:

1. Прийом та обробка сигналів переривання від зовнішніх пристроїв, кількістю, що дорівнює 25 пристроям;
2. Максимальна частота синхронізації дорівнює 2 МГц;
3. Застосування напівпровідникового матричного індикатора розмірністю 8 рядків та 5 стовбців;
4. Частота розгортки зображення на індикаторі дорівнює 100 Гц.

Також було проведено повне структурно-функціональне проектування. Описано процес розробки програмної частини програмно-апаратного комплексу з повним поясненням ключових моментів.

Окрім детального опису процесу розробки програми, було створено пояснення до роботи з програмно-апаратним комплексом. Описано основні головні вікна, кнопки та можливості програмно-апаратного комплексу.

					ЕЛІТ 6.171.00.10.415 ПЗ	Арк.
						45
Зм.	Лист	№ докум.	Підпис	Дата		

СПИСОК ЛІТЕРАТУРИ

1. Питание организованных коллективов в детских садах, школах, КШП, спортивных школах, летних лагерях, санаториях, домах отдыха, интернатах, медицинских учреждениях [Электронный ресурс]. – 2017. – Режим доступа до ресурсу: <https://www.bestnet.ru/programs/pitanie/>.
2. Мова програмування С# [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <http://mycsharp.ru/>.
3. Microsoft Visual Studio [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://visualstudio.microsoft.com/ru/>.
4. Встречайте: Open Server! [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://ospanel.io/>.
5. PIC16F877 Datasheet (PDF) - Microchip Technology [Электронный ресурс] – Режим доступа до ресурсу: <https://pdf1.alldatasheet.com/datasheet-pdf/view/75016/MICROCHIP/PIC16F877.html>.
6. 8259A PROGRAMMABLE INTERRUPT CONTROLLER (8259A/8259A-2). – 24 с
7. Basics of LED dot matrix display. [Электронный ресурс] – Режим доступа до ресурсу: <https://www.radiolocman.com/shem/schematics.html?di=105706>.
8. Розробка функціональної моделі. Методологія IDEF0. URL:https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2.
9. Bringing MySQL to the web [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://www.phpmyadmin.net/>.

					ЕЛІТ 6.171.00.10.415 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		46

ДОДАТОК А

Лістинг програми для роботи мікропроцесорного пристрою на мові програмування PIC-ASM

Програма обробки 25 переривань та вивода зображення на матрицю з 40 індикаторів L835/4HDT в динамічному режимі.

```

list          p=16F877
#include      <p16F877.inc>
__CONFIG    _LVP_OFF & _WDT_OFF & _PWRET_ON & _XT_OSC
; Регістри спеціальних
; функцій
;                                ; АЛГОРИТМ РОБОТИ                                ;
INDF EQU     00H ; Для вводу інформації використана динамічна;
TMRO EQU     01H ; індикація, тобто в конкретний момент ;
PCL EQU      02H ; часу підсвічується тільки 1 стовпець індикатора ;
STATUS EQU   03H ; При перепаді сигналу CLK проходить перемика- ;
FSR EQU      04H ; ння на наступний стовпець. Сигналом RES ;
PORTA EQU    05H ; сбрасивається лічильник стовпців. ;
PORTB EQU    06H ; ;
PORTC EQU    07H ; ;
PORTD EQU    08H ; ;
INTCON EQU   0BH ; ;
OPTREG EQU   081H ; ;
TRISA EQU    085H ; ;
TRISB EQU    086H ; ;
TRISC EQU    087H ; ;
TRISD EQU    088H ; ;
; Регістри ОЗП ; -----;
ADRES EQU    25H ; Адреса переходу на ПП обробки переривання ;
CHR1 EQU     70H ; Код 1 – го символу (усі регістри з адресою 70H ;
CHR2 EQU     71H ; Код 2 – го символу 7FH ;
CHR3 EQU     72H ; Код 3 – го символу ;
CHR4 EQU     73H ; Код 4 – го символу ;
W_TMP EQU    74H ; Тимчасовий регістр ;
CC EQU       7AH ; Лічильник колонок в символі ;
CLK_T EQU    7BH ; Стан сигналу CLK ;
FSR_T EQU    7CH ; Регістр для зберігання FSR ;
C_TEMP EQU   7DH ; Регістр для зберігання адреси матриці символу ;
S_TEMP EQU   7EH ; Регістр для зберігання STATUS ;
P_TEMP EQU   7FH ; Регістр для зберігання PCLATH ;
; Біти регістрів ; ;
RPO EQU      5 ; ;
RP1 EQU      6 ; ;

```

					ЕЛІТ 6.171.00.10.415 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		47

```

A0      EQU      5      ;
Z       EQU      2      ;
RD      EQU      2      ;
WR      EQU      3      ;
GIE     EQU      7      ;
TOIE    EQU      5      ;
INTE    EQU      4      ;
TOIF    EQU      2      ;

INTF    EQU      1      ;
INTA    EQU      2      ;
RES     EQU      0      ;
CLK     EQU      4      ;
;
;
#DEFINE BANK0 BCF STATUS, PRO
#DEFINE BANK1 BCF STATUS, PRO

                ORG 0
                GOTO START
                ORG 4
                GOTO PROCINT

LUT         ADDWF PCL, F
                ORG 9

INT0       GOTO PROCINT0
INT1       GOTO PROCINT1
INT2       GOTO PROCINT2
INT3       GOTO PROCINT3
INT4       GOTO PROCINT4
INT4       GOTO PROCINT5
INT6       GOTO PROCINT6
INT7       GOTO PROCINT7
INT8       GOTO PROCINT8
INT9       GOTO PROCINT9
INT10      GOTO PROCINT10
INT11      GOTO PROCINT11
INT12      GOTO PROCINT12
INT13      GOTO PROCINT13
INT14      GOTO PROCINT14
INT15      GOTO PROCINT15
INT16      GOTO PROCINT16
INT17      GOTO PROCINT17
INT18      GOTO PROCINT18
INT19      GOTO PROCINT19
INT20      GOTO PROCINT20
INT21      GOTO PROCINT21
INT22      GOTO PROCINT22
INT23      GOTO PROCINT23
INT24      GOTO PROCINT24
;*****

```

```

;Для зручності написання програми ;
;команда BCF STATUS, PRO ;
; замінюється на команду BANC0. ;
;відповідно замінюється і друга ;
;команда ORG 0 задає початкову адресу ;
;пам'яті програми ;
;-----;
;ORG 4 задає початкову адресу прог- ;
;рами обробки переривання ;
;***** ;
;таблиця переходів на переривання 0-24;
;32 інструкції ;

```

```

;***** ;

```

```

БЛОК ПІДПРОГРАМ
;*****
; PROCINT Підпрограма обробки переривань (35 інструкції)
;
PROCINT  MOVWF  W_TPM
        SWAPF  STATUS, W
        CLRf   STATUS
        MOVWF  S_TPM
        MOVF   PCLATH, W
        MOVWF  P_TMP
        BTFSS  INTCON, INTF
        GOTO   NOINT
        BANK1
        MOVLW  B`11111111`
        MOVWF  TRISD^80H
        BANK0
        BSF    PORTB, INTA
        BCF    PORTB, INTA
        BSF    PORTB, INTA
        MOVF   PORTD, W
        MOVWF  ADRES
        BCF    PORTB, INTA
        BSF    PORTB, INTA
        BCF    PORTB, INTA
        RRF    ADRES, F
        RRF    ADRES, F
        MOVF   ADRES, W
        CALL   LUT
        GOTO   EXITP
NOINT   BTFSS  INTCON, T0IF
        GOTO   EXITIP
        CALL   SHOW
EXITIP  MOVF   P_TMP, W
        MOVWF  PCLATH
        SWAPF  S_TMP, W
        MOVFW  STATUS
        SWAPF  W_TMP, F
        SWAPF  W_TMP, W
RETIE
;*****
; SHOW – Підпрограма виводу на матрицю з 40 індикаторів L835/4HDT;
; CHR1-CHR4 коди символів для виводу на індикатор
; матриці символів знаходяться в 1-3 банках по зміщенню 20H (80 байт
; в кожному з банків). Цифри мають код 0-9 відповідно, літери
; алфавіту - з 10 по 39. (55 інструкції)
;-----
SHOW    CLRf   W_TMP
SNT     BSF    PORTA,RES
        BCF    PORTA,RES
        CLRf   CLK_T

```

```

; Обробка зовнішніх переривань;
; PORTD усі входи
;
; надсилаємо INTA1 на PIC-i
; знімаємо INTA1 з PIC-ів
; надсилаємо INTA2 на PIC-i
; читаємо адресу переходу на ПП
; та надсилаємо його в ADRES
; знімаємо INTA2 з PIC-ів
; надсилаємо INTA3 на PIC-i
; знімаємо INTA3 з PIC-ів
; ділимо адресу переходу на 4
;
; и зберігаємо його в W
; перехід на обробники переривань;
; переривання по переповненню. TMRO
; вивід на індикатор

```

```

        MOVLW CHR1                ;
    MOVWF FSR                    ;Адреса регістра з кодом 1-го символу
NC      MOVF   FSR, W
        MOVWF FSR_TMP           ; зберігаємо адресу регістра з кодом симв.
        MOVF   INDF, W         ; читаємо код символу в W
        MOVWF C_TMP            ; знаходження банку з матрицею символів
        BTFSC C_TMP, 5
        GOTO   BANK3
        BTFSC C_TMP, 4
        GOTO   BANK2
        BANK1
        GOTO   CALCA
BANK3   BANK1
BANK2   BSF    STATUS, RP1
CALCA   ANDVF 15                ; скидаємо старший півбайт кода символів
        MOVWF C_TMP
        RLF   C_TMP, F
        RLF   C_TMP, F
        ADDWF C_TMP, W         ; знаходження. адреси 1-го байта матриці символів
        CLRF  CC                ; скинули номер колонки в символі
        MOVWF FSR                ; В FSR – адреса в банку 1-го байт
        ; 5-ти байтного образу символів
NCC     MOVF   INDF, W          ; W – байт образу символів
        MOVF   STATUS, W
        MOVWF S_TMP           ; зберегли дані про поточний банк
        BCF   STATUS, RP1
        BANK0                 ; виставили банк 0
        MOVWF PORTC           ; вивели на індикатор стовпець
        BTFSC CLK_T, 0         ; далі перемкнули на наступний стовпець
        GOTO   CLK0
        BSF   CLK_T, 0
        BSF   PORTB, CLK
        GOTO   NEXTC
CLK0    BCF   CLK_T, 0
        BCF   PORTB, CLK
NEXTC   MOVF   S_TMP, W
        MOVWF STATUS           ; відновили робочий банк
        INCF  FSR, F
        INCF  CC, F
        MOVF  CC, W
XORLW  5                        ; вивели усі 5 стовпців?
BTFSS  STATUS, Z
GOTO   NCC                     ; ні – продовжуємо виводити наст. стовпці
MOVF   FSR_TMP, W             ; перемикаємо на наст. символ
MOVWF  FSR
INCF   FSR, R
BTFSS  FSR, 2                 ; вивели усі 4 символи?
GOTO   NC                     ; ні - продовжуємо
INCF   W_TMP, F               ; збільшили лічильник показів
BTFSS  W_TMP, 4               ; показали 16 разів
GOTO   SNT                    ; ні – показіємо ще раз
RETURN ;-----;

```

```

;*****
;*****
;Ділянка ініціалізації: програмування PIC-ов та ін. (104 інструкції) ;
;*****
START BANK1 ;
MOVLW B`00000000` ; PORTA усі виходи ;
MOVWF TRISA^80H ; ;
MOVLW B`00000001` ; PORTB<0> - вхід (INT), решта виходи; ;
MOVWF TRISB^80H ; ;
MOVLW B`00000000` ; PORTC усі виходи ;
MOVWF TRISC^80H ; ;
MOVLW B`00000000` ; PORTD усі виходи ;
MOVWF TRISD^80H ; ;
MOVLW B`00000101` ; Установка дільника перед TMR0, коеф. ;
MOVWF OPTREG^80H ; ділення 1:64 ;
MOVLW B`10110000` ; Установка режиму обробки переривань: ;
MOVWF INTCON ; переповнення TMN0 та INT ;
BANK0 ; ;
MOVLW B`00110100` ; Налаштування 0-го PIC-а: ;
MOVWF PORTD ; Адреса IRQ0-32 ;
MOVLW B`11000001` ; ;
MOVWF PORTA ; ;
BSF PORTA, 2 ; ;
BSF PORTB, 3 ; ICW1 ;
BCF PORTB, 3 ; ;
BCF PORTA, 2 ; ;
BCF PORTA, 5 ; ;
CLRF PORTD ; ;
BSF PORTA, 5 ; ;
BSF PORTA, 2 ; ;
BSF PORTB, 3 ; ICW2 ;
BCF PORTB, 3 ; ;
BCF PORTA, 2 ; ;
BCF PORTA, 5 ; ;
BSF PORTA, 5 ; ;
BSF PORTA, 2 ; ;
BSF PORTB, 3 ; ICW3 ;
BCF PORTB, 3 ; ;
BCF PORTA, 2 ; ;
BCF PORTA, 5 ; ;
MOVLW B`01010100` ; Налаштування 1-го PIC-а ;
MOVWF PORTD ; Адреса IRQ0 – 64 ;
BSF PORTA, 3 ; ;
BSF PORTB, 3 ; ICW1 ;
BCF PORTB, 3 ; ;
BCF PORTA, 3 ; ;
BCF PORTA, 5 ; ;
CLRF PORTD ; ;
BSF PORTA, 5 ; ;
BSF PORTA, 3 ; ;
BSF PORTB, 3 ; ICW2 ;
BCF PORTB, 3 ; ;

```

```

BCF     PORTA, 3      ;
BCF     PORTA, 5      ;
MOVLW  B`00000001`   ;
MOVWF  PORTD          ;
BSF     PORTA, 5      ;
BSF     PORTA, 3      ;
BSF     PORTB, 3      ; ICW3
BCF     PORTB, 3      ;
BCF     PORTA, 3      ;
BCF     PORTA, 5      ;
MOVLW  B`01110100`   ; Налаштування 2-го PIC-а
MOVWF  PORTD          ; Адреса IRQ0 – 96
BSF     PORTA, 4      ;
BSF     PORTB, 3      ; ICW1
BCF     PORTB, 3      ;
BCF     PORTA, 4      ;
BCF     PORTA, 5      ;
CLRF   PORTD          ;
BSF     PORTA, 5      ;
BSF     PORTA, 4      ;
BSF     PORTB, 3      ; ICW2
BCF     PORTB, 3      ;
BCF     PORTA, 4      ;
BCF     PORTA, 5      ;
MOVLW  B`00000010`   ;
MOVWF  PORTD          ;
BSF     PORTA, 5      ;
BSF     PORTA, 4      ;
BSF     PORTB, 3      ; ICW3
BCF     PORTB, 3      ;
BCF     PORTA, 4      ;
BCF     PORTA, 5      ;
MOVLW  B`00010100`   ; Налаштування 3-го PIC-а
MOVWF  PORTD          ; Адреса IRQ0 – 0
BSF     PORTA, 1      ;
BSF     PORTB, 3      ; ICW1
BCF     PORTB, 3      ;
BCF     PORTA, 1      ;
BCF     PORTA, 5      ;
CLRF   PORTD          ;

BSF     PORTA, 5      ;
BSF     PORTA, 1      ;
BSF     PORTB, 3      ; ICW2
BSF     PORTB, 3      ;
BSF     PORTA, 1      ;
BSF     PORTA, 5      ;
MOVLW  B`00000111`   ;
MOVLW  PORTD          ;
BSF     PORTA, 5      ;
BSF     PORTA, 1      ;
BSF     PORTB, 3      ; ICW3

```



```

BSF      PORTB, 3      ;
BSF      PORTA, 1     ;
BSF      PORTA, 5     ;
CLRF     TMRO         ; Обнуляємо TMRO
CALL     MAIN         ; Виклик основної програми
END;

```

ДОДАТОК Б

Перелік елементів

Позначення на схемі	Найменування	Кількість
DD1	PIC16F877-04/L	1
DD2,DD3	SN74LS241N	2
DD4	CD45188	1
DD5	CD4051A	1
DD6,DD7	SN74LS145	2
DD8-DD11	Intel 8259	4
DD12,DD13	SN74LS04	2
C1,C2	K10-19 20пФ 5%	2
C3	K10-19 100 нФ 5%	1
VD1-VD40	L835/4HDT	40
ZQ1	Murata Erie CDA2.00MG	1

ДОДАТОК В

Лістинг програмного коду

meal.sql

```
-- phpMyAdmin SQL Dump
-- version 4.8.3
-- https://www.phpmyadmin.net/
--
-- Хост: 127.0.0.1:3306
-- Время создания: Фев 14 2020 г., 00:50
-- Версия сервера: 8.0.12
-- Версия PHP: 5.5.38

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- База данных: `meal`
--
-----

--
-- Структура таблицы `dish`
--

CREATE TABLE `dish` (
  `dish_id` int(11) NOT NULL,
  `name` varchar(25) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Дамп данных таблицы `dish`
--
```

					ЕЛІТ 6.171.00.10.415 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		55

```
INSERT INTO `dish` (`dish_id`, `name`) VALUES
(1, 'Суп'),
(2, 'Борщ'),
(3, 'Салат'),
(13, 'Каша');
```

```
--
-- Структура таблицы `dish_prod`
--
```

```
CREATE TABLE `dish_prod` (
  `dp_id` int(11) NOT NULL,
  `weight` int(11) NOT NULL,
  `prod_id` int(11) NOT NULL,
  `dish_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
--
-- Дамп данных таблицы `dish_prod`
--
```

```
INSERT INTO `dish_prod` (`dp_id`, `weight`, `prod_id`, `dish_id`) VALUES
(1, 100, 2, 2),
(2, 50, 1, 2);
```

```
--
-- Структура таблицы `meal`
--
```

```
CREATE TABLE `meal` (
  `meal_id` int(11) NOT NULL,
  `name` varchar(25) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
--
-- Дамп данных таблицы `meal`
--
```

```
INSERT INTO `meal` (`meal_id`, `name`) VALUES
(1, 'Обід'),
(3, 'Назва'),
(5, 'New meal'),
(6, 'J,tl 2');
```

```
--
-- Структура таблицы `meal_dish`
```

```
--  
  
CREATE TABLE `meal_dish` (  
  `md_id` int(11) NOT NULL,  
  `count` int(11) NOT NULL,  
  `meal_id` int(11) NOT NULL,  
  `dish_id` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
--  
-- Дамп данных таблицы `meal_dish`  
--
```

```
INSERT INTO `meal_dish` (`md_id`, `count`, `meal_id`, `dish_id`) VALUES  
(1, 2, 1, 2),  
(2, 3, 2, 1),  
(3, 1, 1, 1);
```

```
-----  
  
--  
-- Структура таблицы `product`  
--
```

```
CREATE TABLE `product` (  
  `prod_id` int(11) NOT NULL,  
  `name` varchar(25) NOT NULL,  
  `proteins` float NOT NULL,  
  `fat` float NOT NULL,  
  `carbo` float NOT NULL,  
  `calorie` float NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
--  
-- Дамп данных таблицы `product`  
--
```

```
INSERT INTO `product` (`prod_id`, `name`, `proteins`, `fat`, `carbo`, `calorie`) VALUES  
(1, 'Горох', 20.5, 2, 54.3, 298),  
(2, 'Борошно', 10.6, 1.3, 73.2, 331),  
(3, 'Крупа манна', 10.3, 1, 67.9, 328),  
(4, 'Крупа гречана', 12.6, 3.3, 63.2, 335),  
(5, 'Рис', 7, 1, 71.8, 330),  
(6, 'Пшоно', 11.5, 3.3, 67.2, 348),  
(7, 'Крупа вівсяна', 11, 6.1, 52.5, 303),  
(8, 'Крупа перлова', 9.3, 1.1, 67.5, 320),  
(9, 'Крупа ячна', 10, 1.3, 67.7, 324),  
(10, 'Макаронні вироби', 10.4, 1.1, 69.8, 337),  
(11, 'Хліб житній', 6.6, 1.2, 41.1, 202),  
(12, 'Хліб пшеничний', 7.6, 0.9, 46.9, 231),  
(13, 'Булка міська', 7.8, 2.5, 50.7, 261),  
(14, 'Сухарі', 10.9, 1.5, 69.6, 341);
```

					ЕЛІТ 6.171.00.10.415 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		57

```

--
-- Индексы сохранённых таблиц
--

--
-- Индексы таблицы `dish`
--
ALTER TABLE `dish`
  ADD PRIMARY KEY (`dish_id`);

--
-- Индексы таблицы `dish_prod`
--
ALTER TABLE `dish_prod`
  ADD PRIMARY KEY (`dp_id`);

--
-- Индексы таблицы `meal`
--
ALTER TABLE `meal`
  ADD PRIMARY KEY (`meal_id`);

--
-- Индексы таблицы `meal_dish`
--
ALTER TABLE `meal_dish`
  ADD PRIMARY KEY (`md_id`);

--
-- Индексы таблицы `product`
--
ALTER TABLE `product`
  ADD PRIMARY KEY (`prod_id`);

--
-- AUTO_INCREMENT для сохранённых таблиц
--

--
-- AUTO_INCREMENT для таблицы `dish`
--
ALTER TABLE `dish`
  MODIFY `dish_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=15;

--
-- AUTO_INCREMENT для таблицы `dish_prod`
--
ALTER TABLE `dish_prod`
  MODIFY `dp_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;

--

```

					ЕЛІТ 6.171.00.10.415 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		58

```

-- AUTO_INCREMENT для таблицы `meal`
--
ALTER TABLE `meal`
  MODIFY `meal_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;

--
-- AUTO_INCREMENT для таблицы `meal_dish`
--
ALTER TABLE `meal_dish`
  MODIFY `md_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

--
-- AUTO_INCREMENT для таблицы `product`
--
ALTER TABLE `product`
  MODIFY `prod_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=17;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

add.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
using MySql.Data.MySqlClient;
using System.Data;
namespace meal
{
    /// <summary>
    /// Логика взаимодействия для add.xaml
    /// </summary>
    public partial class add : Window
    {
        bool first_start = true;
        DataTable dt;
        MySqlDataAdapter dataAdopter;
        static String mySqlStr = "server=localhost; user=root;database=meal;password=";
        MySqlConnection sqlConnect = new MySqlConnection(mySqlStr);
        public add()
        {
            InitializeComponent();
        }
        public void UpdateComponents()
        {
            dataGrid.Visibility= Visibility.Hidden;

```

										Арк.
										59
Эм..	Лист	№ докум.	Підпис	Дата	ЕЛІТ 6.171.00.10.415 ПЗ					

```

if (meal.MainWindow.isMeal)
{
    comboLabel.Content = "Оберіть страву";
    boxLabel.Content = "Кількість порцій";
    meinLabel.Content = $"Додати страву на \"{meal.MainWindow.name}\"";
}
else {
    comboLabel.Content = "Оберіть продукту";
    boxLabel.Content = "Вага продукту";
    meinLabel.Content = $"Додати продукт в \"{meal.MainWindow.name}\"";
}

UpdateTable();
for (int i = 0; i < dataGrid.Items.Count - 1; i++)
{
    object[] array;
    DataRowView path = dataGrid.Items[i] as DataRowView;
    array = path.Row.ItemArray;
    addComboBox.Items.Add(array[0].ToString());
}
}
private void firstActivated(object sender, EventArgs e)
{
    if (first_start) { UpdateComponents(); first_start = false; }
}
private void AddBtn_Click(object sender, RoutedEventArgs e)
{
    if (meal.MainWindow.isMeal)
        SqlNonRequestQuery($"INSERT INTO meal_dish( count, meal_id, dish_id)
VALUES ({cont.Text},{meal.MainWindow.idValue},{SELECT dish_id FROM dish WHERE
name=\"{addComboBox.SelectedValue.ToString()}\" limit 1) ");
    else
        SqlNonRequestQuery($"INSERT INTO dish_prod( weight, prod_id, dish_id)
VALUES ( {cont.Text}, (SELECT prod_id FROM product WHERE
name=\"{addComboBox.SelectedValue.ToString()}\" limit 1), {meal.MainWindow.idValue})");
    Close();
}
}
void SqlNonRequestQuery(string sqlq)
{
    try
    {
        sqlConnect.Open();
        MySqlCommand createCommand = new MySqlCommand(sqlq, sqlConnect);
        createCommand.ExecuteNonQuery();
        sqlConnect.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Помилка підключення до бази даних {ex.Message}");
        Close();
    }
}
void UpdateDataTable(DataTable d, String sqlq)
{
    sqlConnect.Open();
    MySqlCommand createCommand = new MySqlCommand(sqlq, sqlConnect);
    dataAdopter = new MySqlDataAdapter(createCommand);
    dataAdopter.Fill(d);
    sqlConnect.Close();
}
}
void UpdateTable()
{
    try
    {
        dt = new DataTable();

```

									Арк.
									60
Зм.	Лист	№ докум.	Підпис	Дата	ЕЛІТ 6.171.00.10.415 ПЗ				


```

        UpdateDataTable(dt, Sqlquery());
        dataGrid.ItemsSource = dt.DefaultView;
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Помилка підключення до бази даних {ex.Message}");
        Close();
    }
    string Sqlquery()
    {
        if (meal.MainWindow.isMeal) return "SELECT distinct name FROM dish;";
        else return "SELECT distinct name FROM product;";
    }
}
private void Cancel_Click(object sender, RoutedEventArgs e)
{
    Close();
}
}
}

```

MainWindow.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using System.Globalization;
using System.Windows.Media.Animation;
using MySql.Data.MySqlClient;
using System.Data;

using Word = Microsoft.Office.Interop.Word;

namespace meal
{
    /// <summary>
    /// Логика взаємодії для MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        DataTable dt;
        MySqlDataAdapter dataAdopter;
        MySqlCommandBuilder cmdbl;
        static String mySqlStr = "server=localhost; user=root;database=meal;password=";
        MySqlConnection sqlConnect = new MySqlConnection(mySqlStr);
        Word.Application app;
        bool first_start = true;
        public static string idValue = "-1";
        public static string name = "";
    }
}

```

									Арк.
									61
Зм.	Лист	№ докум.	Підпис	Дата	ЕЛІТ 6.171.00.10.415 ПЗ				

```

String header="";
public static bool isMeal =true;
int elementsQ = 0;

public MainWindow()
{
    InitializeComponent();
}
string Sqlquery()
{
    String sqlQueryStr = "";
    switch (selectList.SelectedIndex) {
        case 0:
            {
                sqlQueryStr = "select meal_id, name as 'Назва',
ifnull(proteins,0) as 'Білки', " +
                "ifnull(fat,0) as 'Жир', ifnull(carbo,0) as 'Вугливоди' , " +
                "ifnull(calories,0) as 'Калорії' from meal as mealt " +
                "left join (SELECT round(avg( meal_id),0) as m_id, " +
                "sum(weight*count) as weight, sum(proteins*count) as
proteins, " +
                "sum(fat*count) as fat, sum(carbo*count) as carbo, " +
                "sum(calories*count) as calories from meal_dish as m left
join" +
                " (SELECT round(avg(a.dish_id),0) as d_id, name,
round(sum(ifnull(weight,0)),2) as weight, " +
                "round(sum(ifnull(proteins,0)),2) as proteins,
round(sum(ifnull(fat,0)),2) as fat, " +
                "round(sum(ifnull(carbo,0)),2) as carbo,
round(sum(ifnull(calorie,0)),2) as calories " +
                "from dish as a left join (SELECT dish_id, weight, " +
                "(select proteins/100*weight from product where
product.prod_id=dish_prod.prod_id) as proteins, " +
                "(select fat/100*weight from product where
product.prod_id=dish_prod.prod_id) as fat, " +
                "(select carbo/100*weight from product where
product.prod_id=dish_prod.prod_id) as carbo, " +
                "(select calorie/100*weight from product where
product.prod_id=dish_prod.prod_id) as calorie " +
                "FROM dish_prod) as my_tb on my_tb.dish_id= a.dish_id group
by a.name) as m_d " +
                "on m.dish_id=m_d.d_id group by meal_id) as allt on meal_id =
m_id";
                break;
            }
        case 1:
            {
                sqlQueryStr = "SELECT round(avg(a.dish_id),0) as d_id, name as
'Назва'," +
                " round(sum(ifnull(weight,0)),2) as 'Вага' ," +
                " round(sum(ifnull(proteins,0)),2) as 'Білки'," +
                "round(sum(ifnull(fat,0)),2) as 'Жири', " +
                "round(sum(ifnull(carbo,0)),2) as 'Вуглеводи', " +
                "round(sum(ifnull(calorie,0)),2) as 'Калорії' from dish as a
" +
                "left join (SELECT dish_id, weight, " +
                "(select proteins/100*weight from product where
product.prod_id=dish_prod.prod_id) as proteins, " +
                "(select fat/100*weight from product where
product.prod_id=dish_prod.prod_id) as fat," +
                "(select carbo/100*weight from product where
product.prod_id= dish_prod.prod_id) as carbo, " +

```

```

        "(select calorie/100*weight from product where
product.prod_id=dish_prod.prod_id) as calorie FROM dish_prod) " +
        "as my_tb on my_tb.dish_id= a.dish_id group by a.name;";
    break;
    }
    case 2:
    {
        sqlQueryStr = "SELECT prod_id, name as 'Назва', proteins as
'Білки', fat as 'Жири', carbo as 'Вуглеводи', calorie as 'Калорії' FROM product;";
        break;
    }
    default:
    {
        if (isMeal)
        {
            sqlQueryStr = $"SELECT md_id, " +
                $"(select name from meal where
meal.meal_id=meal_dish.meal_id) as 'Назва прийому їжі' , " +
                $"count as 'Кільсть', (select name from dish where
dish.dish_id=meal_dish.dish_id) as 'Назва страви'" +
                $"FROM meal_dish WHERE meal_id={idValue}";
        } else {
            sqlQueryStr = $"SELECT dp_id,(select name from dish where
dish.dish_id=dish_prod.dish_id) as 'Назва страви', " +
                $"weight as 'Вага', (select name from product where
product.prod_id=dish_prod.prod_id) as 'Назва продукту' " +
                $"FROM dish_prod where dish_id={idValue}";
        }
        break;
    }
    }
    return sqlQueryStr;
}
void UpdateWindow() {
    UpdateComponents();
    UpdateTable();
}

void UpdateComponents() {
    if (selectList.SelectedIndex != -1)
        h1.Content = selectList.SelectedValue.ToString().Substring(38);
    else h1.Content = header;
    dataGrid.CanUserAddRows = (selectList.SelectedIndex == 2);
    dataGrid.CanUserDeleteRows = (selectList.SelectedIndex == 2);
    dataGrid.IsReadOnly = !(selectList.SelectedIndex == 2);
    Next.Visibility = Visibility.Hidden;
    dellDish.Visibility = Visibility.Hidden;
    editBTN.Visibility = Visibility.Hidden;
    addDish.Visibility = Visibility.Visible;
    forProducts.Visibility = Visibility.Hidden;
    forMeal.Visibility = Visibility.Hidden;
    forDish.Visibility = Visibility.Hidden;
    temp.Visibility = Visibility.Hidden;
    prod.Visibility = Visibility.Hidden;

    if (selectList.SelectedIndex ==2 || selectList.SelectedIndex ==-1) {
        dishName.Visibility = Visibility.Hidden;
    } else {
        dishName.Visibility = Visibility.Visible;
    }

    if (selectList.SelectedIndex == -1){
        Save.Visibility = Visibility.Hidden;
    } else {
        Save.Visibility = Visibility.Visible;
    }
}

```

						ЕЛІТ 6.171.00.10.415 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата			63

```

    }

    if (selectList.SelectedIndex == 2) {
        Reset.Visibility = Visibility.Visible;
        Commit.Visibility = Visibility.Visible;
    } else {
        Reset.Visibility = Visibility.Hidden;
        Commit.Visibility = Visibility.Hidden;
    }
}

void UpdateDataTable(DataTable d, String sqlq){
    sqlConnect.Open();
    MySqlCommand createCommand = new MySqlCommand(sqlq, sqlConnect);
    dataAdopter = new MySqlDataAdapter(createCommand);
    dataAdopter.Fill(d);
    sqlConnect.Close();
} void UpdateTable()
{
    try {
        dt = new DataTable();
        UpdateDataTable(dt, Sqlquery());
        dataGrid.ItemsSource = dt.DefaultView;
        dataGrid.Columns[0].Visibility = Visibility.Hidden;
        dt.Columns[0].AutoIncrement = true;
    }
    catch (Exception)
    {
        MessageBox.Show($"Помилка підключення до бази даних");
        Close();
    }
}

private void Activ_(object sender, EventArgs e)
{
    if (first_start) { UpdateWindow(); first_start = false; }
}

private void selectListClosed(object sender, EventArgs e)
{
    UpdateWindow();
    if (selectList.SelectedIndex == 2)
        elementsQ = dataGrid.Items.Count-1;
}

private void Commit_Click(object sender, RoutedEventArgs e)
{
    try
    {
        if (dataGrid.Items.Count-1 == elementsQ)
        {
            for (int i = 0; i < elementsQ; i++) {
                object[] array;
                DataRowView path = dataGrid.Items[i] as DataRowView;
                array = path.Row.ItemArray;
                String str = $"UPDATE product SET
name=\"{array[1].ToString()}\"$proteins={ array[2].ToString()}$fat={
array[3].ToString()}$carbo={ array[4].ToString()}$calorie={ array[5].ToString()} WHERE
prod_id = { array[0].ToString()}";
                h1.Content = str.Replace(',', '.').Replace('$', ',');
                SqlNonRequestQuery(str.Replace(',', '.').Replace('$', ','));
            }
        } else {
            cmdbl = new MySqlCommandBuilder(dataAdopter);

```

						Арк.
						64
Зм.	Лист	№ докум.	Підпис	Дата	ЕЛІТ 6.171.00.10.415 ПЗ	

```

        dataAdopter.Update(dt);
        UpdateTable();
        MessageBox.Show("Дані успішно оновлено!");
    }
    UpdateWindow();
}
catch (Exception ex) {
    MessageBox.Show(ex.Message);
}
}

private void Reset_Click(object sender, RoutedEventArgs e)
{
    Next.Visibility = Visibility.Hidden;
    UpdateWindow();
}

private void CurrentCell(object sender, EventArgs e)
{
    if (selectList.SelectedIndex != 2 ){
        Next.Visibility = Visibility.Visible;
        dellDish.Visibility = Visibility.Visible;
    }
    if (selectList.SelectedIndex == -1){
        Next.Visibility = Visibility.Hidden;
        editBTN.Visibility = Visibility.Visible;
        dellDish.Visibility = Visibility.Visible;
    }
}
private void dataGridLostFocus(object sender, RoutedEventArgs e) { }

private void nextClick(object sender, RoutedEventArgs e)
{
    DataRowView path = dataGrid.Items[dataGrid.SelectedIndex] as DataRowView;
    object[] array = path.Row.ItemArray;
    idValue = array[0].ToString();
    name = array[1].ToString();
    isMeal = selectList.SelectedIndex == 0;
    if (selectList.SelectedIndex == 0)
        header = "Всі страви на " + array[1].ToString().ToLower();
    else header = "Всі продукти страви " + array[1].ToString().ToLower();
    selectList.SelectedIndex = -1;
    UpdateWindow();
}

private void AddDishClick(object sender, RoutedEventArgs e)
{
    switch (selectList.SelectedIndex)
    {
        case 0: { SqlNonRequestQuery($"INSERT INTO meal (name) VALUES
(\\"{dishName.Text}\\");"); break; }
        case 1: { SqlNonRequestQuery($"INSERT INTO dish (name) VALUES
(\\"{dishName.Text}\\");"); break; }
        case -1: {
            add addwindow = new add();
            addwindow.ShowDialog();

            break; }
    }

    UpdateWindow();
}

private void DellDish(object sender, RoutedEventArgs e)
{
    DataRowView path = dataGrid.Items[dataGrid.SelectedIndex] as DataRowView;
    object[] array = path.Row.ItemArray;

```

									Арк.
									65
Зм.	Лист	№ докум.	Підпис	Дата	ЕЛІТ 6.171.00.10.415 ПЗ				

```

        switch (selectList.SelectedIndex)
        {
            case 0: { SqlNonRequestQuery($"DELETE FROM meal WHERE
meal_id={array[0].ToString()}"); break; }
            case 1: { SqlNonRequestQuery($"DELETE FROM dish WHERE
dish_id={array[0].ToString()}"); break; }
            case -1: {
                if (isMeal){ SqlNonRequestQuery($"DELETE FROM meal_dish WHERE
md_id={array[0].ToString()}");
                } else {      SqlNonRequestQuery($"DELETE FROM dish_prod WHERE
dp_id={array[0].ToString()}"); }
                break; }
        }
        UpdateWindow();
    }

    void SqlNonRequestQuery(string sqlq) {
        sqlConnect.Open();
        MySqlCommand createCommand = new MySqlCommand(sqlq, sqlConnect);
        createCommand.ExecuteNonQuery();
        sqlConnect.Close();
    }

    private void Resize_wind(object sender, SizeChangedEventArgs e)
    {
        Wind.Width = 767;
        Wind.Height = 459;
    }
    public static string editID;
    public static string editName;
    public static string editCount;
    public static string editName2;
    private void edit_click(object sender, RoutedEventArgs e)
    {
        DataRowView path = dataGrid.Items[dataGrid.SelectedIndex] as DataRowView;
        object[] array = path.Row.ItemArray;
        editID = array[0].ToString();
        editName = array[1].ToString();
        editCount = array[2].ToString();
        editName2 = array[3].ToString();

        Edit edit = new Edit();
        edit.ShowDialog();
        UpdateWindow();
    }

    private void saveMenu_Click(object sender, RoutedEventArgs e) {
        app = new Word.Application();
        app.Visible = false;
        var oDoc = app.Documents.Add(@"C:\Users\1\Desktop\temp.dotx");
        UpdateTableSave(forProducts, "SELECT prod_id, name FROM product where prod_id
in (select prod_id from dish_prod where dish_id in (select dish_id from meal_dish))");
        //UpdateTableSave(forProducts, "SELECT prod_id, name FROM product;");
        var p = oDoc.Content.Paragraphs.Add();
        p.Range.Text = "ЗАТВЕРДЖУЮ: ГОЛОВНИЙ ЛІКАР _____\t\t\tМЕНЮ
ПРОДУКТІВ НА _____\n";

        object oMissing = System.Reflection.Missing.Value;
        object oEndOfDoc = "\\endofdoc";

        Word.Table newTable;
        Word.Range wrdRng = oDoc.Bookmarks.get_Item(ref oEndOfDoc).Range;
        newTable = oDoc.Tables.Add(wrdRng, 1, forProducts.Items.Count - 1 + 8, ref
oMissing, ref oMissing);
        newTable.Borders.InsideLineStyle = Word.WdLineStyle.wdLineStyleSingle;
    }

```

										Арк.
										66
Зм.	Лист	№ докум.	Підпис	Дата						

```

newTable.Borders.OutsideLineStyle = Word.WdLineStyle.wdLineStyleSingle;
newTable.AllowAutoFit = true;
newTable.Cell(1, 1).Range.Text = "Меню";
newTable.Cell(1, 2).Range.Text = "Кількість";
newTable.Cell(1, 2 + forProducts.Items.Count).Range.Text = "Б";
newTable.Cell(1, 2 + forProducts.Items.Count + 1).Range.Text = "Ж";
newTable.Cell(1, 2 + forProducts.Items.Count + 2).Range.Text = "В";
newTable.Cell(1, 2 + forProducts.Items.Count + 3).Range.Text = "Калорії";
newTable.Cell(1, 2 + forProducts.Items.Count + 4).Range.Text = "Вихід";
newTable.Cell(1, 2 + forProducts.Items.Count + 5).Range.Text = "Підпис";

for (int i = 0; i < forProducts.Items.Count - 1; i++) {
    newTable.Cell(1, i + 3).Range.Text = (forProducts.Items[i] as
DataRowView).Row.ItemArray[1].ToString());
}
UpdateTableSave(forMeal, "SELECT meal_id, name FROM meal;");
int counter = 1;
for (int i = 0; i < forMeal.Items.Count - 1; i++){
    object[] array;
    DataRowView path = forMeal.Items[i] as DataRowView;
    array = path.Row.ItemArray;
    double[] summ= new double[forProducts.Items.Count];
    double[] dopsumm = { 0, 0, 0, 0 };
    for (int k = 0; k < forProducts.Items.Count - 1; k++)
        summ[k] = 0;
    newTable.Rows.Add(); counter++;
    newTable.Cell(counter, 1).Range.Text = array[1].ToString();
    String query = "SELECT round(avg(a.dish_id),0) as d_id, name, " +
"round(sum(ifnull(weight,0)),2) as weight
,round(sum(ifnull(proteins,0)),2) as proteins, " +
"round(sum(ifnull(fat,0)),2) as fat, round(sum(ifnull(carbo,0)),2) as
carbo," +
"round(sum(ifnull(calorie,0)),2) as calories from dish as a " +
"left join (SELECT dish_id, weight, " +
"(select proteins/100*weight from product where
product.prod_id=dish_prod.prod_id) as proteins, " +
"(select fat/100*weight from product where
product.prod_id=dish_prod.prod_id) as fat," +
"(select carbo/100*weight from product where product.prod_id=
dish_prod.prod_id) as carbo, " +
"(select calorie/100*weight from product where
product.prod_id=dish_prod.prod_id) as calorie FROM dish_prod) " +
"as my_tb on my_tb.dish_id= a.dish_id " +
$"where a.dish_id in (select dish_id from meal_dish where meal_id =
{array[0].ToString()}) group by a.name;";
    UpdateTableSave(forDish, query);
    for (int j = 0; j < forDish.Items.Count - 1; j++) {
        object[] dishArray;
        DataRowView dishPath = forDish.Items[j] as DataRowView;
        dishArray = dishPath.Row.ItemArray;
        newTable.Rows.Add(); counter++;
        newTable.Cell(counter, 1).Range.Text = dishArray[1].ToString();

        UpdateTableSave(temp, $"SELECT count FROM meal_dish WHERE meal_id
={array[0].ToString()} and dish_id = {dishArray[0].ToString()}");
        newTable.Cell(counter, 2).Range.Text = (temp.Items[0] as
DataRowView).Row.ItemArray[0].ToString());
        newTable.Cell(counter, 6 + forProducts.Items.Count).Range.Text =
dishArray[2].ToString();
        newTable.Cell(counter, 2 + forProducts.Items.Count).Range.Text =
dishArray[3].ToString();
        newTable.Cell(counter, 3 + forProducts.Items.Count).Range.Text =
dishArray[4].ToString();
        newTable.Cell(counter, 4 + forProducts.Items.Count).Range.Text =
dishArray[5].ToString();
    }
}

```

									Арк.
									67
Зм.	Лист	№ докум.	Підпис	Дата					

```

        newTable.Cell(counter, 5 + forProducts.Items.Count).Range.Text =
dishArray[6].ToString();
        dopsumm[0] += Convert.ToDouble(dishArray[3].ToString())*
Convert.ToDouble((temp.Items[0] as DataRowView).Row.ItemArray[0].ToString());
        dopsumm[1] += Convert.ToDouble(dishArray[4].ToString())*
Convert.ToDouble((temp.Items[0] as DataRowView).Row.ItemArray[0].ToString());
        dopsumm[2] += Convert.ToDouble(dishArray[5].ToString())*
Convert.ToDouble((temp.Items[0] as DataRowView).Row.ItemArray[0].ToString());
        dopsumm[3] += Convert.ToDouble(dishArray[6].ToString())*
Convert.ToDouble((temp.Items[0] as DataRowView).Row.ItemArray[0].ToString());
        for (int k = 0; k < forProducts.Items.Count - 1; k++) {
            UpdateTableSave(prod, $"SELECT IFNULL((select weight FROM
dish_prod WHERE prod_id = {(forProducts.Items[k] as
DataRowView).Row.ItemArray[0].ToString()} and dish_id={ dishArray[0].ToString()}),0) as w
from dish_prod limit 1 ");
            if (Convert.ToDouble((prod.Items[0] as
DataRowView).Row.ItemArray[0].ToString()) > 0.001) {
                newTable.Cell(counter, k + 3).Range.Text = (prod.Items[0] as
DataRowView).Row.ItemArray[0].ToString();
                summ[k] += Convert.ToDouble((temp.Items[0] as
DataRowView).Row.ItemArray[0].ToString())* Convert.ToDouble((prod.Items[0] as
DataRowView).Row.ItemArray[0].ToString());
            }
        }
    }
    newTable.Rows.Add(); counter++;
    newTable.Cell(counter, 1).Range.Text = "Всього";
    for (int k = 0; k < forProducts.Items.Count - 1; k++) {
        if(summ[k]>0.001)
            newTable.Cell(counter, k + 3).Range.Text = summ[k].ToString() ;
    }
    for (int k = 0; k < 4; k++) {
        newTable.Cell(counter, k + 2+ forProducts.Items.Count).Range.Text =
dopsumm[k].ToString();
    }

    }
    var p2 = oDoc.Content.Paragraphs.Add();
    p2.Range.Text = "\n\t\tДИЕТЦЕНТРА _____
КУХАР _____ \n\t\tДИЕТОЛОГ _____ ШЕФ-
КУХАР _____ \t\tБУХГАЛТЕР _____ ";

    try
    {
        oDoc.Save();
        oDoc.Close();
    }
    catch (System.Runtime.InteropServices.COMException) { }
    app.Quit();
}

void UpdateDataTableSave(DataTable d, String sqlq)
{
    sqlConnect.Open();
    MySqlCommand createCommand = new MySqlCommand(sqlq, sqlConnect);
    dataAdopter = new MySqlDataAdapter(createCommand);
    dataAdopter.Fill(d);
    sqlConnect.Close();
}

void UpdateTableSave(DataGrid dg, string sqlq)
{
    try {
        dt = new DataTable();

```

						Арк.
						68
Зм.	Лист	№ докум.	Підпис	Дата	ЕЛІТ 6.171.00.10.415 ПЗ	


```

        UpdateDataTableSave(dt, sqlq);
        dg.ItemsSource = dt.DefaultView;
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Помилка підключення до бази даних {ex.Message}");
        Close();
    }
}

String[] mealArr = { "№", "Прийом їжі", "Білки", "Жири", "Вуглеводи", "Калорії"
};
String[] dishArr = { "№", "Страва", "Білки", "Жири", "Вуглеводи", "Калорії" };
String[] prodArr = { "№", "Продукт", "Білки", "Жири", "Вуглеводи", "Калорії" };

private void saveClick(object sender, RoutedEventArgs e)
{
    int count = (selectList.SelectedIndex == 2) ? (dataGrid.Items.Count - 1) :
(dataGrid.Items.Count);
    if (count != 0)
    {
        UpdateTable();
        app = new Word.Application();
        app.Visible = false;
        var oDoc = app.Documents.Add();
        var p = oDoc.Content.Paragraphs.Add();
        p.Range.Text = $"{1.Content}";
        object oMissing = System.Reflection.Missing.Value;
        object oEndOfDoc = "\\endofdoc";
        Word.Table newTable;
        Word.Range wrdRng = oDoc.Bookmarks.get_Item(ref oEndOfDoc).Range;
        newTable = oDoc.Tables.Add(wrdRng, 1, dataGrid.Columns.Count, ref
oMissing, ref oMissing);
        newTable.Borders.InsideLineStyle = Word.WdLineStyle.wdLineStyleSingle;
        newTable.Borders.OutsideLineStyle = Word.WdLineStyle.wdLineStyleSingle;
        newTable.AllowAutoFit = true;

        for (int k = 0; k < dataGrid.Columns.Count; k++)
        {
            switch (selectList.SelectedIndex)
            {
                case 0: { newTable.Cell(1, 1 + k).Range.Text = mealArr[k]; break;
}
                case 1: { newTable.Cell(1, 1 + k).Range.Text = dishArr[k]; break;
}
                case 2: { newTable.Cell(1, 1 + k).Range.Text = prodArr[k]; break;
}
                default: { break; }
            }
        }

        for (int i = 0; i < count; i++)
        {
            object[] array;
            try
            {
                newTable.Rows.Add();
                DataRowView path = dataGrid.Items[i] as DataRowView;
                array = path.Row.ItemArray;
                newTable.Cell(i + 2, 1).Range.Text = (i + 1).ToString();
                for (int j = 1; j < dataGrid.Columns.Count; j++)
                    newTable.Cell(i + 2, 1 + j).Range.Text = array[j].ToString();
            }
            catch (NullReferenceException) { break; }
        }
    }
}

```

										Арк.
										69
Зм.	Лист	№ докум.	Підпис	Дата	ЕЛІТ 6.171.00.10.415 ПЗ					

```

        try
        {
            oDoc.Save();
            oDoc.Close();
        }
        catch (System.Runtime.InteropServices.COMException) { }
        app.Quit();
    }
    else
    {
        MessageBox.Show("Insufficient data!");
    }
}
}
}

```

Edit.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
using MySql.Data.MySqlClient;
using System.Data;

namespace meal
{
    /// <summary>
    /// Логика взаємодії для Edit.xaml
    /// </summary>
    public partial class Edit : Window
    {
        bool first_start = true;
        static String mySqlStr = "server=localhost; user=root;database=meal;password=";
        MySqlConnection sqlConnect = new MySqlConnection(mySqlStr);

        public Edit()
        {
            InitializeComponent();
        }

        public void UpdateComponents()
        {
            name.Text = meal.MainWindow.editName2;
            count.Text = meal.MainWindow.editCount;
            mainLabel.Content = $"Вміст \"{meal.MainWindow.editName2}\" в
            \"{meal.MainWindow.editName}\"";
            if (meal.MainWindow.isMeal)
            {

```

									Арк.
									70
Зм.	Лист	№ докум.	Підпис	Дата	ЕЛІТ 6.171.00.10.415 ПЗ				

```

        nameLabel.Content = "Страва";
        countLabel.Content = "Кількість порцій";
    }
    else
    {
        nameLabel.Content = "Продукт";
        countLabel.Content = "Вага";
    }
}

void SqlNonRequestQuery(string sqlq)
{
    try
    {
        sqlConnect.Open();
        MySqlCommand createCommand = new MySqlCommand(sqlq, sqlConnect);
        createCommand.ExecuteNonQuery();
        sqlConnect.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Помилка підключення до бази даних {ex.Message}");
        Close();
    }
}

private void firstActivated(object sender, EventArgs e)
{
    if (first_start) { UpdateComponents(); first_start = false; }
}

private void Button_Click(object sender, RoutedEventArgs e)
{
    Close();
}

private void Button_Click_1(object sender, RoutedEventArgs e)
{
    if (meal.MainWindow.isMeal) {
        SqlNonRequestQuery($"UPDATE meal_dish SET count={count.Text} WHERE md_id
= {meal.MainWindow.editID};");
    }
    else {
        SqlNonRequestQuery($"UPDATE dish_prod SET weight={count.Text} WHERE dp_id
= {meal.MainWindow.editID};");
    }
    Close();
}
}
}
}

```

					ЕЛІТ 6.171.00.10.415 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		71