

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних систем управління

Бакалаврська дипломна робота

на тему:

«Автоматизована система корекції приймальної гільзи у протезуванні»

Виконав: студент 4_курсу, групи АКІТ-22мс
спеціальності 151 – Автоматизація та
комп'ютерно-інтегровані технології

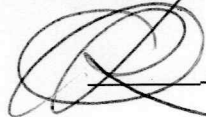
 Віталій СТУКЕЛЬМАН

Керівник: ст. викл. каф. КСУ

 Василь ПРИСЯЖНЮК

« 12 » червня 2024 р.

Рецензент: к.т.н., доц., каф. АІТ

 Костянтин ОВЧИННИКОВ

« 12 » червня 2024 р.

Допущено до захисту
Зав. кафедри КСУ

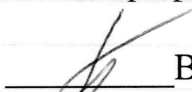
 В'ячеслав Ковтун
« 14 » червня 2024р.

Вінниця ВНТУ – 2024рік

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних систем управління
Рівень вищої освіти перший (бакалаврський)
Галузь знань – 15 – Автоматизація та приладобудування
Спеціальність – 151 – Автоматизація та комп'ютерно-інтегровані технології
Освітньо - професійна програма – Інтелектуальні комп'ютерні системи управління

ЗАТВЕРДЖУЮ

Завідувач кафедри КСУ
д.т.н., проф.


В'ячеслав КОВТУН
“24” травня 2024 року

**ЗАВДАННЯ
НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ**
студенту Стукельману Віталію Сергійовичу.

1. Тема роботи. Автоматизована система корекції приймальної гільзи у протезуванні
керівник роботи ст. викл. каф КСУ Присяжнюк Василь Васильович
затверджені наказом ВНТУ № 80 від 11 березня .2024 р.
2. Термін подання студентом роботи 12 червня 2024 року
3. Вихідні дані до роботи: Конструювання та технології виготовлення протезів нижніх кінцівок: навч. посіб. / А. Д. Салєєва, О. Г. Аврунін, П. О. Баєв, С. В. Корнєєв, Я. В. Носова, І. В. Кабаненко, М. В. Зайцев, Т. О. Трофименко, І. Л. Тимофєєв. - Харків: ХНУРЕ, 2023. - 481 с
Гільзи та системи кріплення протезів нижніх кінцівок URL https://langs.physio-pedia.com/uk/lower-limb-prosthetic-sockets-and-suspension-systems-uk/#cite_note-8-2 (дата звернення: 23.05.24)
В.М. Дубовой. Методичні вказівки до виконання бакалаврських/бакалавирських робіт для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології». Вінниця :ВНТУ,2022.URL <https://iq.vntu.edu.ua/repository/card.php?id=5929&lang=uk>
4. Зміст текстової частини: вступ, аналіз об'єкта автоматизації, вибір та обґрунтування системи, розробка програмного забезпечення, програмне забезпечення, висновки, перелік посилань, додатки
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень) ERM-діаграма бази даних Структура бази даних в середовищі phpMyAdmin Структура таблиці “form_description Вигляд заповненої даними

таблиці “form_description”. Структура таблиці “main”. Структура таблиці “measurements Вигляд вікна “Таблиці звітності” створеного додатку Вигляд вікна результатів роботи програми

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдан ня	виконан ня

7. Дата видачі завдання 27.05.2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
	Вибір, узгодження та затвердження теми БДР	06.05.2024	08.05.2024	
	Постановка задачі дослідження	17.05.2024	19.05.2024	
	Аналіз об'єкта автоматизації, робота з літературою	20.05.2024	26.05.2024	
	Розробка та налаштування програмного забезпечення, підбір архітектури та інструментів розробки	27.05.2024	03.06.2024	
	Оформлення пояснювальної записки, графічного матеріалу	04.06.2024	08.06.2024	
	Попередній захист БДР, доопрацювання, рецензування БДР	09.06.2024	10.06.2024	
	Захист БДР	20.06.2024	20.06.2024	

Студент _____
(підпис)

Віталій СТУКЕЛЬМАН
(ім'я і ПРИЗВИЩЕ)

Керівник роботи _____
(підпис)

Василь ПРИСЯЖНЮК
(ім'я і ПРИЗВИЩЕ)

АНОТАЦІЯ

Бакалаврська дипломна робота складається з 72 сторінок формату А4, на яких є 15 рисунків, 4 додатка, список використаних джерел містить 20 найменувань.

Метою дослідження є підвищення ефективності системи корекції приймальної гільзи у протезуванні, проводячи аналіз існуючих систем автоматизованого виготовлення та корекції в протезуванні та розробляючи власний програмний продукт, що може дозволити автоматизувати процес контролю та обліку виробництва протезів.

У теоретично-методичній частині роботи розглянуто та проаналізовано предметну область та об'єкт автоматизації системи корекції приймальної гільзи у протезуванні, її основні принципи, елементи автоматизованої системи і визначено оцінки ефективності функціонування облікових систем. Визначено основні переваги та недоліки автоматизованої системи. У практичній частині виконана розробка та проведене тестування системи для надання послуг обліку контролю процесу виготовлення протезів.

Ключові слова: автоматизація, облікові системи, етапи виготовлення протезів, автоматизована інформаційна система, підприємство, програмний застосунок.

ABSTRACT

The bachelor thesis consists of 72 pages of A4 format, on which there are 15 figures, 4 appendices, the list of used sources contains 20 names.

The purpose of the study is to increase the effectiveness of the system of correction of the receiving sleeve in prosthetics by analyzing the existing systems of automated manufacturing and correction in prosthetics and developing its own software product that can allow automating the process of controlling and accounting for the production of prosthetics.

In the theoretical-methodical part of the work, the subject area and the object of the automation of the receiving sleeve correction system in prosthetics, its main principles, elements of the automated system, and the evaluation of the effectiveness of the accounting systems are considered and analyzed. The main advantages and disadvantages of the automated system are determined. In the practical part, the development and testing of the system for the provision of accounting and control services for the manufacturing process of prostheses was carried out

Keywords: automation, accounting systems, stages of manufacturing prostheses, automated information system, enterprise, software application

.

ЗМІСТ

ВСТУП	3
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	5
1.1 Актуальність необхідності розробки.....	6
1.2 Аналіз проблемних питань.....	11
1.3 Постановка задачі.....	23
2 РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ.....	28
2.1 Вимоги до нової системи.....	28
2.2 Розробка бази даних системи.....	29
3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	36
3.1 Функціонал програми.....	36
3.2 Вибір мови програмування.....	37
3.3 Вибір редактора коду та його особливості.....	38
3.4 Опис розробленої програм.....	39
ВИСНОВКИ.....	46
Перелік посилань.....	47
Додатки	49
ДОДАТОК А (Обов'язковий) Протокол перевірки на наявність запозичень.....	50
ДОДАТОК Б (обов'язковий) Технічне завдання.....	51
ДОДАТОК В (обов'язковий) Ілюстративна частина.....	54
ДОДАТОК Г (довідковий) Лістинг програми Prosthetic Manager ClientCare Pro Prosthetic Database Manager.....	65

ВСТУП

Актуальність. Проблема забезпечення травмованих людей сучасними, якісними, багатофункціональними протезами верхньої та нижньої кінцівки завжди була актуальною. В нашій державі, після повномасштабного вторгнення агресора, надзвичайно збільшилась потреба в протезуванні. Виготовлення протезів включає в себе перелік різноманітних задач, юридично-правові, фінансові, логістика, технологічні, технічні, організаційні, тому для своєчасного та якісного виготовлення протезів і необхідна автоматизована система.

Метою дослідження є підвищення ефективності системи корекції приймальної гільзи у протезуванні, проводячи аналіз існуючих систем автоматизованого виготовлення та корекції в протезуванні та розробляючи власний програмний продукт, що може дозволити автоматизувати процес контролю та обліку виробництва протезів.

Для досягнення поставленої мети необхідно розв'язати наступні **задачі**:

- провести аналіз предметної області та об'єкту автоматизації;
- дослідити програмні засоби для автоматизованих виготовлення та корекції в протезуванні;
- розробка програмного забезпечення моніторингу процесу виготовлення протезів

Об'єктом дослідження є процес автоматизованого виготовлення та корекції в протезуванні, який проводиться розробленою системою та включає в себе комплекс процесів, методів та інструментів, що використовуються для ефективного управління усіма складовими частинами процесу виготовлення та корекції виготовлення протезів.

Предметом дослідження є автоматизована система корекції приймальної гільзи у протезуванні. Основними етапами дослідження є аналіз інформаційних систем, розробка математичної моделі системи, проектування клієнтської частини, вибір технологій для програмної реалізації та представлення результатів роботи програмного рішення.

Інновація розробленої системи полягає у створенні власного програмного рішення за допомогою сучасних інноваційних технологій.

Практична цінність полягає в розробці програмного забезпечення, яке дозволяє автоматизувати систему корекції приймальної гільзи у протезуванні, ефективно використовувати ресурси підприємства, дозволяє контролювати всі етапи виготовлення, своєчасно вносити корективи в технологічний процес, моніторити наявність та терміни постачання компонентів та вузлів, контролювати терміни та якість замовлення, фіксувати недоліки та побажання клієнтів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність необхідності розробки

Необхідність надання допомоги людям, у яких ампутовані кінцівки є в даний час є важливою проблемою для людства. Тільки в сполучених штатах в допомозі потребують майже два мільйони людей живуть із втратою кінцівок. Щороку в США відбувається приблизно 185000 ампутацій через причини, що варіюються від судинних захворювань до травм. Більше того, протезування може коштувати від 5000 до 50 000 доларів.

За понад півтора роки масштабної війни в Україні близько 50 тисяч українців стали інвалідами. Вони втратили руки чи ноги. Про це повідомляє американське видання Wall Street Journal.

Аналітики повторюють оцінку Ottobock, найбільшого у світі виробника протезів, та його медичних партнерів.

Слід зазначити, що фактично кількість людей з інвалідністю внаслідок ампутації кінцівки може перевищувати 50 тисяч. Фактом є те, що процес протезування вимагає часу. Ампутація часто виконується через кілька тижнів або навіть місяців після травми.

Київський фонд надії підрахував:

- під час війни було важко поранено близько 200 тис. українців;
- близько 20 000 з них отримали поранення, які вимагали ампутації.

«Для порівняння, приблизно 67 000 німців і 41 000 британців ампутували кінцівки під час Першої світової війни, коли ця процедура часто була єдиним способом уникнути смерті». Менш ніж 2000 американських ветеранів вторгнення в Афганістан та Ірак мають ампутації", - пише американське видання.

У статті WSJ йдеться, що фактична кількість ампутованих може бути більшою, оскільки протезування потребує часу, а деякі жертви чекають на ампутацію місяцями після поранення.

За словами українських лікарів і спеціалізованих клінік, система охорони здоров'я перевантажена, і багато пацієнтів чекають нового члена більше року. За словами керівника мережі центрів медичної реабілітації «Unbroken» Олександра Кобзарєва, тільки у Львові минулого року лікарі провели понад 53 тисячі операцій.

Щоб задовольнити значно зростаючий попит на протезування, на український ринок вийшли багато іноземних компаній, а також відкривають свої двері нові вітчизняні протезні компанії.

Протезування складається з багатьох невід'ємних, послідовних і додаткових частин, різних методів виробництва, виробничих процесів і процесу реабілітації. Висновки реабілітації, спрямовані на підвищення їх якості, зручності, комфорту використання, суттєво впливають на способи і способи виготовлення протезів.

За складом, як функціонально завершений виріб для застосування, складається з безлічі різних деталей і вузлів, початок комбінування деталей і вузлів поклав в 1919 році Отто Бок у своїй компанії «Orthopädische Industrie», який застосував поділ протезів на окремі елементи: нога, збірка «коліно-гомілка» та індивідуальна приймальна манжета [2]. Таким чином, у процесі виготовлення протезів було закладено важливу основу для їх систематизації та вдосконалення. Даний винахід, технічне рішення, яке і зараз використовуються для виробництва протезів, надало можливість розпочати промислове виробництво протезів (ноги, колінні суглоби, шини тощо).

Збірка протезів здійснюється згідно з інструкцією з монтажу з урахуванням анатомічних особливостей і особливостей пацієнта разом із ствольною коробкою, виготовленою на замовлення. Конструкція протезів має два типи, які суттєво відрізняються один від іншого, а саме оболонкові (немодульні) і модульні протези. Також протези нижніх кінцівок поділяються на типи за типом ампутованої кінцівки, а саме: протези стегна, протези великогомілкової кістки, протези ніг і плавальні протези.

Протези стегна. Протези модульного типу, застосовуються при ампутації

на рівні стегна виготовляються (рис.1.1). Основні частини таких протезів : куксоприймальної гільзи, гільзового адаптера, колінного вузла, модуля та стопи. Куксоприймальні гільзи, зазвичай, забезпечуються вакуумним кріпленням.



Рисунок 1.1 – Будова протезу стегна

Для встановлення протеза використовується еластичний бинт або спеціальна подовжена кришка, яка витягується з отвору в гільзі, потім за допомогою клапана виштовхується повітря і міцно фіксується гільза за допомогою вакууму. [3]

Щоб зняти протез, достатньо відкрити клапан і витягнути ніжку. Протез також можна закріпити на куксі за допомогою кліпси, еластичного бинта або силіконової вставки.

Протезування гомілки. При ампутації на рівні гомілки використовують протези (рис. 1.2). Виготовляються модульні протези. Модульний протез великогомілкової кістки складається з ніжкоприймаючої втулки, вкладки для

лапки, модуля і лапки. Якщо кукса гомілки занадто коротка або в колінному суглобі є патології, її також можна встановити за допомогою манжети (манжети).

Вали приймання кукси для протезів великогомілкової кістки можуть мати різну конструкцію. по всій поверхні колінної чашечки.



Рисунок 1.2 – Будова протезів гомілки та стегна

Протезування стопи. Після ампутації на рівні стопи протези потрібні в основному в косметичних цілях. Людина Лісфранка або Шопарда з ампутованими кінцівками може стояти або ходити по дому без допомоги протеза, але під час ходьби з протезом ці пацієнти можуть повністю відчувати опорну поверхню, що дає їм відчуття безпеки.

За даними Саїма і Пирогова, при протезуванні після ампутацій кінцівка, що приймає куксу, зазвичай починається трохи нижче коліна. Пацієнтам, які використовують цей вид протезування, може знадобитися спеціальне взуття з амортизацією під здорову стопу, щоб компенсувати різницю в довжині кінцівок.

Купальні протези призначають при ампутаціях на рівні гомілки і стегна для прийняття душу, користування водою, пересування у водоймах.

Протез гомілки, призначений для плавання, повністю виготовлений з ламінованого пластику. Може мати технологічні отвори для зменшення плавучості у воді. Такі протези оснащені тільки ніжками без суглобів.

Для плавання протез кульшового суглоба оснащений вологостійкими стопорними петлями і безшарнірними ніжками. Фіксація протеза під час купання зазвичай вакуумна або поперекова.

Ці елементи протезів є унітарними елементами і виробляються на великих підприємствах, які використовують власну автоматизовану систему виготовлення елементів протезів.

У конструкції протезів передбачена індивідуальна приймальна рукоятка (рис. 1.3), яка за своїми розмірами та конфігурацією є унікальним, індивідуальним елементом. Його особливість полягає в тому, що це єдиний елемент протеза, який стикається куксою з тілом людини.



Рисунок 1.3 – Види приймальної гільзи

Куксоприймальна гільза, або приймальна гільза - елемент протеза, виготовлений індивідуально і в прямому контакті з тілом людини, сприймаючи вагу пацієнта. Найбільш поширеним є приймальне крило з акрилових смол [5]. Може мати натуральний колір кузова або індивідуальний дизайн за бажанням замовника. Також можлива скелетонована версія з внутрішньою гнучкою ствольною коробкою.

Дата 25.04.2024

Бланк замірів № 4

П.І.Б. Гречаник Олександр Михайлович

Адреса

Вид виробу Шифр ПН.6.2.3.4.0 Найменування протез стегна і колінного суглоба

Шифр

Найменування

Рівень мобільності ☐ 1 рівень ☐ 2 рівень ☒ 3 рівень ☐ 4 рівень

Стать ☒ чоловік ☐ жінка

Вік 25

Зріст 175

Вага 78

Об'ємний розмір та довжина куки

ураження
лівої
кістки
(стегно)

Комплектуючі

Висота каблук

Гілеза, адаптер (ротатійний), колінний механізм, стопа

Розмір стопи, см

Рисунок 1.4 – Бланк замовлення на виготовлення протезу стегна

Послідовність підготовки приймальної руки така: зняття антропометричних параметрів кінцівки пацієнта (рис. 1.4), створення

приймальної руки за обраною або запропонованою технологією, складання, корекція та адаптація руки. характеристика форми тулуба пацієнта, після чого починається складання протеза з окремих частин.

Якість і точність її виготовлення є надзвичайно важливими характеристиками індивідуальної приймальної гільзи, що суттєво впливає на якість зібраного протезу. З метою виділення важливості, акцентування уваги на якість виготовлення цієї деталі і дана назва даній бакалаврській роботі.

1.2 Аналіз проблемних питань

Розробка автоматизованої системи є складний та довгий процес. Необхідно враховувати існуючі фактори, обмеження, режими робота та експлуатації, та багато інших факторів та особливостей. На мій погляд однією із головних проблем при розробці систем, які надають керівництву, персоналу та користувачам, клієнтам різноманітну інформацію, а також відслідковують, контролюють термін виконання різних етапів виробництва і замовлення в цілому, є вірна та точна побудова бази для зберігання інформації, тобто структура бази даних.

Нова база даних, яка буде розроблена для даної системи повинна в своїй структурі враховувати всі етапи виготовлення, корекції та збірки протезів. Також, бажано, щоби база, а в подальшому і програмне забезпечення, мали можливість до розширення своїх можливостей та функцій, тобто задача стоїть таким чином, щоби база і програмне забезпечення могли виконувати функції які не передбачені вимогами до даної системи, а наперед, якби планували можливі розширення на майбутнє. Для правильної побудови бази даних, яка би враховувала би всі операції по виготовлення протезів, спочатку треба провести аналіз типів, різновидів та технологій виготовлення, визначитися з головними етапами виробництва, з'ясувати можливість взаємозаміни матеріалів та елементів різних видів протезів, тому проведемо аналіз існуючих видів протезів та технологій їх виготовлення.

В даний час існують дві системи протезів, які поділяються за конструкцією та компонуванням – протези оболонкового (немодульного) та модульного типу:

Протези каркасного (оболонкового) виконання, їх називають також традиційними або екзоскелетними, немодульними. Такі протези виготовляють переважно з дерева або пластмаси [7]. Стінки протезів здійснюють як формоутворювальну, так і несучу функцію. Приймальні гільзи немодульних протезів найчастіше робляться зі шкіри, дерева. Хоча застосовується і ливарна смола. Немодульні протези мають вкрай обмеженою функціональністю, тому підходять тільки для протезування літніх і неактивних пацієнтів.

Приймальні гільзи немодульних протезів найчастіше виготовляють зі шкіри, дерева. Хоча застосовується і ливарна смола.

Немодульні протези мають вкрай обмежену функціональність, тому підходять тільки для протезування літніх і неактивних пацієнтів.

Модульні протези, їх називають також трубчастими або ендоскелетними. У цих протезах несучу функцію здійснює трубчаста конструкція. Зовнішню форму їм надає гнучке косметичний покриття зі спіненого матеріалу.

Модульні протези перевершують немодульні (каркасні) в функціональності і можливості регулювань. Поряд зі взаємозамінністю вузлів модульні протези забезпечують можливість установки в них додаткових вузлів, що підвищують функціональні якості протезів. Модульна система була представлена в 1969 році як трубчастий кістковий протез з косметичної пенопластової облицюванням; і продовжує постійно вдосконалюватися [9]. Найважливішими передумовами для реалізації такої концепції стали перш за все побажання пацієнтів, відповідно до яких були встановлені наступні критерії конструкції протезів:

- відновлення природного зовнішнього вигляду втраченої кінцівки;
- підвищення зручності користування;
- збільшення безпеки в процесі експлуатації;

- розширення функціональних можливостей.

Головні критерії конструкції протезів:

- косметичність, відновлення натурального виду втраченої кінцівки;
- підвищення зручності користування;
- збільшення безпеки в процесі експлуатації;
- розширення функціональних можливостей.

Приймальна гільза [11]. Для виготовлення приймальних гільз протезів стегна використовують ливарні смоли, термопластичні матеріали, дерево, інші. Може застосовуватися і чохол полімерний гелевий або силіконовий.

Полімерні силіконові чи гелеві чохла не тільки підвищують зручність носіння протеза, а й забезпечують:

- оптимальний розподіл навантаження по всій поверхні кукси;
- гарне демпфірування ударів;
- пом'якшення в області кісткових виступів і чутливих ділянок;
- згладжування нерівностей кукси;
- зниження поршневих рухів і зменшення потенційних ушкоджень шкіри;
- надійне кріплення на культі за рахунок зчеплення з шкірою і застосування замкового механізму.

Вимоги та побудова гільзи протезів. Приймальна гільза є найважливішим індивідуальним модулем протезу. У складі протезу приймальна гільза об'єднує усічену кінцівку з механічними модулями і сприймає основні статичні та динамічні навантаження в системі «людина-протез».

Основними функціями протеза є комфортне розміщення кукси в рукаві, поглинання навантаження під час стояння та ходьби, передача зусиль для керування протезом та утримання його на куксі. Під час роботи протеза приймальна втулка чинить розмірене навантаження на тканини кульки, компенсує ударні навантаження, мінімізуючи переміщення поршня кукси в приймальній порожнині втулки і не перешкоджаючи нормальному кровообігу

в закритій кінцівці.

Раціональна конструкція приймальних стулок протезів кульшового суглоба, правильний вибір їх форми, досить точне встановлення умов посадки кукси, правильний контроль дають змогу отримати добрі результати протезування.

Існують різні системи та форми приймальних гільз протеза кульшового суглоба. Серед основних типів крил, які відрізняються формою посадкового кільця, виділяють поперечно-овальну (квадратну) і поздовжньо-овальну (САТ САМ) конфігурації. Рецепторні рукави можуть бути твердими, які зберігають форму, і м'якими (шкірястими), які потім деформуються. Невідповідність конфігурації стовбура та ствольної коробки може спричинити бічну нестабільність.

Форма крила визначає передачу сил і рухів. Таким чином, відмінності у формах крил призводять до відмінностей у дії зовнішніх навантажень, що передаються від протеза на куксу, і в особистих відчуттях пацієнта з ампутацією цих навантажень.

При розробці ствольної коробки необхідно враховувати анатомічні, фізіологічні, антропометричні та біомеханічні аспекти. Зокрема, перш за все, варто звернути увагу на певну особливість будови нижніх кінцівок людини. Дефекти конструкції типових ствольних рукояток найбільше проявляються у складних і нетипових протезах нижніх кінцівок.

Останнім часом технологія протезування нижніх кінцівок стає все більш індивідуальною. Індивідуальність протезів поєднує в собі кілька елементів, кожен з яких істотно впливає на результат:

- індивідуальне виготовлення приймального плеча протеза.
- індивідуальний підбір елементів протеза;
- особа, що відповідає характеристикам опорно-легеневого апарату тулуба, схема конструкції протеза;

Якість протеза залежить від форми руки. Протез руки повинен відповідати вимогам [12].

1. Міцний зв'язок між тулубом і рукою. Кожен «помилковий суглоб», «мертвий простір», навіть невеликий, ускладнює управління протезом, знижує його надійність при стоянні і ходьбі, збільшує енерговитрати.

2. Не залишайте локального тиску на окремі частини стовбура і не залишайте інші ненавантаженими;

3. Повний контакт. Хороший контакт всієї поверхні руки з куксою необхідний для надійного з'єднання кукси з протезом руки. Не повинно бути зон холостого ходу. Кожна частина тулуба повинна стикатися, тобто відчувати легку напругу. Але якщо такий контакт нестерпний для кукси, то перед встановленням протезів необхідно поставити питання про поліпшення якості кукси.

4. Навантаження на торцеву поверхню кукси. Забезпечення максимальної підтримки периферійної поверхні дозволяє протезу наблизитися до природного механізму руху. Спираючись на кінцівки тулуба, пацієнт отримує важливу непряму інформацію про навантаження на землю та властивості землі через сенсорну нервову систему. Величина навантаження на кінцеву поверхню залежить від рівня ампутації, якості м'яких тканин і техніки операції. Торцева поверхня кукси може бути навантажена рано, якщо рубець знаходиться за межами зони навантаження, зазвичай вперед.

5. Циркуляція та іннервація. Для забезпечення повного тісного надійного контакту і навантаження на кінцеву поверхню, намагаються не порушувати артеріальний, венозний або лімфатичний кровообіг та іннервацію. Цього можна досягти, якщо поперечний переріз проксимального отвору великий. Занадто вузький вхідний отвір рукава рано чи пізно неминуче призведе до хронічного стазу та набряку кінчика кукси. Наслідком цього є підвищене потовиділення, утворення пухирів внаслідок перегріву та хронічні виразки, значні зміни шкіри тулуба та основи руки. Торцева поверхня стовбура втрачає здатність витримувати навантаження, зусилля перерозподіляються через вужчу вхідну частину крила і контур замикається. Рука не повинна здавлювати судини і нерви тулуба, уникати виникнення венозного застою, травми, карієсу,

наміну і бурситу.

6. Введення та видалення протеза руки не повинно вимагати великих фізичних і розумових зусиль або сторонньої допомоги. Конструкція протеза повинна забезпечувати його надягання і зняття в зручному і надійному положенні тіла, зокрема в положенні сидячи.

7. Вимірювання зовнішніх розмірів. Протез не повинен бути більшим за живу нижню кінцівку. З естетичної точки зору навіть краще, якщо протез буде трохи вужчим за іншу ніжку.

8. Не допускати повороту приймальної руки відносно тулуба, не обмежувати рухів тазостегнового суглоба при ходьбі та не ускладнювати сидіння хворого, не створювати обмежень під час фізіологічних виходів та особистого догляду.

9. Догляд. Протез повинен бути простим і легким в обслуговуванні. Особливо це стосується внутрішніх поверхонь. Необхідно уникати неприємного запаху. Цю проблему вирішує протез руки, який можна мити зсередини. З твердим зовнішнім покриттям гладка внутрішня поверхня полегшує щоденний догляд.

10. Подальша корекція протеза. Розміри кукси зазвичай змінюються в перші місяці після операції, а також пізніше, коли змінюється вага пацієнта. Необхідність досягнення повного контакту між культею та гніздом протеза робить припасування більш частим, ніж раніше.

11. Надійність і довговічність. Протез, навіть при значних навантаженнях, повинен служити довго. З жорсткою втулкою є ризик втрати гладкої внутрішньої поверхні. Навіть маленькі пухирі можуть пошкодити шкіру. Нерівності внутрішньої поверхні необхідно визначати систематичним і своєчасним промацуванням.

При протезуванні нижніх кінцівок протезистам необхідно вирішити такі основні проблеми:

- повністю і правильно розмістити весь об'єм кукси в плечі протеза;
- забезпечити зчеплення кукси протеза з плечем;

- передача статичних навантажень від протеза до скелета;
- забезпечити можливість пересування по протезу.

Повне і правильне розміщення кукси в руці є необхідною умовою для вирішення останніх трьох завдань. Якщо вдасться безболісно і природним шляхом повністю вставити куксу в руку, то перше і друге завдання можна вважати виконаними.

Технології виготовлення крила протеза. Сучасна технологія виготовлення стрілок в Україні передбачає зняття гіпсового негативу з кукси, виготовлення гіпсової моделі кукси, з'єднання та виготовлення постійної гільзи. Використання цієї технології для виготовлення ствольних крил створює низку проблем, які зрештою можуть призвести до того, що початкова форма гільзи буде відрізнятися від форми культі.

Використання гіпсу-негативу та гіпсу-позитиву. Виготовлення гіпсу-негативу. Для виготовлення гіпсу-негативу використовують гіпсові бинти промислового виготовлення шириною 8-10 см, які накладають на куксу в 4-5 шарів. Бинтування проводиться знизу вгору. Спочатку формується дно негативу. Перед гіпсування йому необхідно встановити всі місця, чутливі до тиску [14].

Перед виготовленням гіпсового негатива (зліпка) перевіряють правильність обраного силіконового чохла. Надягають силіконовий чохол на куксу і загвинчують з'єднувальний стержень, на який натягують запобіжний ковпачок. Перед виконанням операцій безпосереднього виготовлення негативу з гіпсових бинтів, щоб уникнути пошкоджень силіконового чохла, надягнутого на куксу, і в якості захисного розділового шару на нього наносять шар тонкої плівки, не перетягуючи м'які тканини.

Поверх нанесеної плівки маркером розмічають проекції власної зв'язки надколінника, головки маломілкової кістки і хворобливих ділянок кукси. Від проекції власної зв'язки надколінника заміряють периметри кукси через кожні 2 см. Потім готують, змочують у воді і накладають на дистальну частину

кукси з надітим силіконовим чохлом чотиришарову гіпсову лонгету, в центрі якої попередньо роблять отвір для з'єднувального стрижня. Від торця кукси в проксимальному напрямку циркулярно наносять 5-6 турів гіпсового бинта.

Ретельно моделюють зліпок з помірним тиском великими пальцями рук в проекції власної зв'язки надколінника. На задній поверхні пальцями обох рук підтискають литкового м'яза. Під час бинтування культя утримується в зігнутому положенні: при культі гомілки в середній третині - під кутом 10-15 °, у верхній третині - під кутом 15-20 °, в нижній третині - під кутом 5-10 °. Намічають кордони країв приймальні гільзи і середину власної зв'язки надколінника.

Одночасно з накладенням гіпсового бинта проводять формування пелота. При цьому моделюють внутрішні ділянки, найбільш витривалі до навантаження. Для цього великим пальцем руки, із зовнішньої частини кукси натискають в потрібних областях з метою формування опорної площадки на поверхні негативу. У такому положенні рука утримується до затвердіння гіпсу.

Виготовлений таким чином негатив видаляють з кукси. Для цього в проекції сухожиль згиначів з обох сторін роблять два лінійних розрізи.

Після схоплювання гіпсу по задній поверхні негативу в проекції сухожиль згиначів гомілки роблять ножицями або ножом вертикальні насічки. При згинанні кукси в колінному суглобі обережно, з обертовими рухами знімають негатив. На отриманому негативі по задньому верхньому краю роблять вертикальні насічки з формуванням відвалів назовні. Обрізають краю гіпсового негатива за формою приймальні гільзи, опустивши задню стінку на 1-1,5 см нижче власної зв'язки надколінника.

Після чого вільний край чохла загортають на негатив. Для видалення негативу культю згинають. Обережними обертальними рухами негатив знімають з кукси. Після зняття негативу на внутрішній його поверхні хімічним олівцем відзначають положення контактної опорного елемента і підсилюють, відбиті з кукси (попередньо наносяться олівцем на куксу) контури кісткових виступів. Чохол разом з прокладками видаляють.

Наступним етапом проводять обробку негатива [8]. Для остаточного моделювання негатив зсередини обробляють тальком, контактнo-опорний елемент відповідно до розмітки знову ставлять на місце. Надягають тонкий сухий чохол, останній натягують, на кінці кукси на чохол наносять губною помадою мітку, і в зігнутому положенні на культю обережно надягають негатив. Після чого просять інваліда зігнути колінний суглоб до 90° і спертися кінцем негативу в ногу техника, імітуючи навантаження в протезі.

Виготовлений таким чином негатив висушують. При відсутності зіткнення кінця кукси з опорно-контактним елементом (про що вказує відсутність відбитка помади на ньому) продовжують підгонку негативу і повторну примірku його до отримання позитивного результату. При незадовільному результаті (не з'являється контакту культі із опорним елементом) необхідно заново виготовити негатив.

Виготовлення гіпсового позитиву. Перед виготовленням позитиву опорноконтактний елемент і прокладки видаляють, підбирають металевий стрижень, встановлюють його в центрі негативу. Заливку гіпсом проводять за загальноприйнятою методикою. Після затвердіння гіпсу негатив розрізають і витягають виготовлений позитив. При обробці позитиву додатковим шаром гіпсу товщиною 3 мм посилюють навантаження по посадковому кільці та інших необхідних місцях.

Гіпсовий позитив обробляється так само звичайним чином:

- області, які повинні бути виключені з контакту - нарощуються гіпсом
- особлива увага приділяється області гребеня великогомілкової кістки, переднього відділу кукси, п'яти і т.д.
- зліпок приводиться у відповідність зі знятими розмірами.

Далі гіпс позитив шліфують, щоб згладити нерівності. На остаточному етапі обробки поверхня позитива загладжується металевою сіточкою. Перед виготовленням приймальні гільзи позитив необхідно повністю висушити.

В першу чергу виготовляють гіпсову гільзу. Далі, після того, як вона повністю висихає, зміцнюють ділянку її з'єднання з іншою частиною протеза.

Саме тривалий період висихання (близько доби) вважається основним недоліком гіпсової гільзи. Тільки після закінчення цього часу вона повністю готова до навантажень.

Використовуючи таку технологію виготовлення приймальних гільз виникає ряд проблем, які можуть призвести до відмінності форми гільзи від культі в кінцевому результаті. Дуже важливо, щоб шар гіпсових бинтів дозволив точно відтворити положення сідничної кістки і зафіксувати отриману форму, що може бути порушено накладанням занадто товстого або занадто тонкого шару гіпсових бинтів. Неправильне положення пальців при моделюванні призводить в подальшому до нестабільної посадки кукси в гільзі, виникненню ротаційних моментів і болісних відчуттів в цій зоні. Під час створення гіпсової моделі кукси можливе неправильне вирівнювання моделі та надання їй неправильної форми.

Техніка литьової смоли в протезуванні. У протезобудування цей метод починає займати панівне становище в 50-і роки. Для виготовлення використовують спеціальні смоли, які розігріваються і з них моделюється протез. Його принцип подібний до виготовлення армованого залізобетону. Залізо надає цементу бажану міцність. Один або кілька шарів волокнистої тканини, посиленої вуглецевими і / або скляними волокнами, служать арматурної основою і заливаються рідкою штучною смолою.

За допомогою вибору і компонування тканини і складу литьової смоли вдається варіювати міцність протеза. Сучасні вуглецеві волокна дозволяють зробити стінку гільзи протеза товщиною в 1-2 мм і тим самим поліпшити зовнішній вигляд і зменшити масу протеза [10].

Для техніки литьової смоли спочатку потрібно зняти гіпс-негатив з кукси. Культю бинтують гіпсовими бинтами таким чином, що знімається зліпок, який в точності відтворює контури кукси.

Ще до висихання його можна заповнити рідким гіпсом. Сюди ж вгіпсовують квадратний профіль з алюмінію, щоб фіксувати лещата навколо гіпс-позитиву. Поки гіпс-позитив висохне, гіпс-негатив видаляють і

залишається точна копія кукси з гіпсу. Але це ще не все. Там, де культя відчуває більш сильне навантаження, гіпс зішліфовують і, навпаки, тонким шаром наносять там, де тиск буде найменшим. На закінчення поверхня відкоригованого позитиву гладко шліфується. Після цього на гіпс-позитив в вакуумі наноситься гільза з литтєвої смоли. Замість епоксидних смол сьогодні стають все поширенішими акрилові смоли.

Створення гільзи протезу за допомогою термопластичних матеріалів. На зміну техніці литтєвої смоли приходить використання термопластів. Вони можуть бути отримані, як правило, з поліетилену, рідше з поліпропілену, так само як пінопласт з закритими осередками.

Завдяки нагріванню матеріал стає пластичним. За допомогою вакуумної обробки можна протягом декількох хвилин на гіпс-позитиві виготовити гільзу з термопласту. Цю технологію застосовують як для прозорих пробних гільз, так і для створення прийомних гільз з м'якими стінками.

Наразі широке розповсюдження отримав спеціальний пластичний матеріал під назвою ТЕРМОЛІН.

Для виготовлення гільзи протезу, її закріплюють на верстаті, розігрівають ТЕРМОЛІН до 180 градусів. Він гріється 40 хвилин. Коли ТЕРМОЛІН провис, ми включають повітря. Під гільзою йде повітря, і на неї натягують ТЕРМОЛІН акуратно на позитив гільзи. Тобто відбувається створення оболонки. Він остигає, а далі отримують примірювальну приймальню гільзу з Термоліну, яка повторює всі анатомічні особливості гіпсової гільзи [7].

Далі в термолінову гільзу вставляють вкладиш, до неї прикручують інші деталі і частини і віддають на примірку пацієнтові. На цьому етапі важливо правильно підігнати термолінову гільзу. Якщо людину щось не влаштовує, її переробляють. Нагрівають ТЕРМОЛІН монтажним феном, матеріал стає пластичним і тоді протезисти надають незручним місцях потрібну форму.

Якщо людині максимально комфортно при примірці, техніки переходять до створення постійної гільзи на основі примірочної.

Гільзу з Термоліну обертають гіпсовим бинтом. По суті, протезист повторює попередній етап - створює зліпок для заливки фінішного варіанту. Але в цьому випадку позитивом для основної гільзи служить термоліновий зразок. У ТЕРМОЛІН заливають гіпсовий розчин. Термолін розрізають і знімають, гіпсовий зразок залишається. Його знову кріплять на верстат і накладають шари тканини упереміш зі спеціальною смолою, яка допомагає фінішній гільзі набувати тверду форму. Але і тут є свої особливості. Перед тим як приступити до намотування тканини, на гіпсовий зліпок надягають спеціальний панчіх, потім фолієвий рукав, а далі накладають 12 шарів трикотажу упереміш зі скловолокном і карбоновими вставками для міцності. Зверху ставлять РСУ - пелюсткову опору замку культеприймача, до якої будуть кріпитися комплектуючі: металева заміна кістки, шарніри стопи і сама стопа.

Зверху на гільзу під вакуумом надягають ще один фолієвий рукав і через отвір зверху заливають суміш з клею, акрилу і косметичної фарби тілесного відтінку. Гільзу залишають на 40 хвилин, за цей час вона твердне, і допрацьовують: зрізають непотрібне, підпилюють потрібне, роблять безпечними краю. Протез майже готовий. Залишається тільки зібрати кінцівку для фінішної примірки ноги.

Використання CAD/CAM технологій для виготовлення протезів. В останні роки велике значення набуло комп'ютерне моделювання та виготовлення індивідуальних приймальних гільз.

При цьому розміри кукси пацієнта використовують в якості бази для фрезерування моделі. Технік-ортопед може за допомогою програмноматематичного забезпечення змодельовати на персональному комп'ютері (ПК) необхідну форму гільзи для конкретного пацієнта. За отриманими даними виготовляють термопластичних пробну гільзу. При остаточному виготовленні приймальні гільзи використовують термопластик або шаруватий пластик на основі литьової смоли.

При знятті мірок використовують спеціальні лазерні сканери. Для

створення гільзи протезу використовують друк на 3D-принтері. Сьогодні цей метод є самим дорогим, але він дозволяє економити час на створення та підгонку протезу.

Технологія використання поліпропілену, розроблена Міжнародним комітетом Червоного Хреста (МКЧХ), застосовується в протезуванні по всьому світу, особливо в країнах з дефіцитом ресурсів та у проєктах МКЧХ. Поліпропілен також використовується в США як альтернатива ламінуванню постійної гільзи. Це дешевше, ніж ламінування, але може виявитися не менш міцною конструкцією.

Система модульних гільз Modular Socket System, розроблена компанією Össur, може бути використана для виготовлення гільзи безпосередньо на кінцівці пацієнта. Це більш простий та швидкий процес, але його вартість значно вища. Термін передачі виробу пацієнту може становити один день. Наразі ця пропозиція доступна для пацієнтів з транстибіальною ампутацією [7].

Більш складною та дорогою технікою, яка використовується сьогодні, є система CAD (Computer-Aided Design, “проєктування з використанням комп’ютерних технологій. Все більшої популярності набувають регульовані гільзи, наприклад, RevoFit2.

1.3 Постановка задачі

В даний час на ринку України, працюють багато різноманітних, як іноземних та і вітчизняних підприємств по виготовленню протезів: компанія Ottobock та «Orto lux», Центр протезно-ортопедичної допомоги ПФ «Теллус», ТОВ «Всеукраїнський центр реабілітації та протезування «ЗДОРОВ’Я», Державне комерційне підприємство "Вінницьке експериментальне протезно-ортопедичне підприємство (ДКП "ВЕПОП") та багато інших.

Також широко застосовуються сучасні методи в іноземних фірмах, які ґрунтуються на застосуванні так званих CAD / CAM / CAE-систем.

Абревіатуру CAD (Computer-Aided Design) використовується для позначення систем автоматизованого проектування, які використовують комп'ютерні технології. Як правило, САПР використовуються для двовимірного та тривимірного геометричного моделювання та проектування машинобудівних виробів, а також для конструкторської документації в програмному середовищі.

CAM (Computer-Aided Manufacturing) — автоматизована система автоматизації виробництва. САМ-системи реалізують автоматизацію вирішення геометричних задач у технологічних процесах, передачу інформації та управління обладнанням і пристроями, необхідними для виробництва деталей.

CAE (Computer Aided Engineering) – системи інженерного аналізу чисельних розрахунків з використанням комп'ютерних технологій, які, як правило, базуються на використанні методу скінченних елементів.

Усі ці процеси доповнюють один одного і зазвичай реалізуються в інтегрованих системах CAD/CAM або CAD/CAM/CAE, які є інноваційними, науково обґрунтованими продуктами, що використовуються в багатьох наукових дисциплінах, особливо в медицині.

Використання систем CAD/CAM/CAE дозволяє скоротити витрати часу та коштів на проектування, планування та виробництво складних інженерних деталей, підвищити якість проектування та технічний рівень результатів, зменшити витрати на масштабне моделювання. і тестування. .

CAD / CAM / CAE – система передбачає наявність певної кількості компонентів, зокрема: технічне забезпечення (електронно-обчислювальне обладнання, телекомунікаційні системи, обладнання для виготовлення деталей, кероване цифровими програмами: 3D-принтери, фрезерні верстати тощо), програмне забезпечення (набір програм). що готують: можливі геометричне моделювання та інженерні розрахунки, зберігання та передачу інформації на пристрій, що забезпечує виготовлення деталей), окремі класи завдань) тощо. [8].

В медицині САПР використовуються для наступних цілей:

- • створення комп'ютерних моделей біологічних об'єктів для діагностики та планування оперативних втручань;
- • відтворення оперативного втручання на віртуальній моделі та прогнозування його анатомічних наслідків;
- • створення моделей дентальних і щелепно-лицевих імплантатів і виробів медичного призначення, вибір їх оптимального розташування з урахуванням топографо-анатомічних співвідношень в області оперативного втручання.

Для виробництва використовуються системи САМ.

- - надруковані на 3D біологічні об'єкти, які можна використовувати для передопераційних вимірювань та проведення «холодної хірургії» на моделі;
- - індивідуальні імпланти, ендопротези, вироби медичного призначення з різних матеріалів.
- - типові рекомендації щодо встановлення зубних імплантатів та щелепно-лицевої хірургії..

Основи, а точніше теоретичні основи автоматизованого проектування для виробництва були розроблені в 1960-1970-х роках. Використання систем CAD/CAM в стоматології почалося в 1980-х роках. в галузі ортопедичної стоматології (С.І. Вольвач, 2002; С.В. Чиканов, 2002). В основі ранніх стоматологічних систем CAD/CAM була технологія виготовлення конструкцій зубних протезів за допомогою комп'ютерного моделювання та фрезерування на машинах, обладнаних цифровим програмним забезпеченням. Вперше такі системи були запропоновані незалежними групами американських і швейцарських учених. Р.В. Андерсон розробив систему ProCERA в 1983 році. В.Х. Моermann і М. Brandestin розробили систему CEREC в Цюрихському університеті в 1985 році, а систему DentiCAD було представлено в 1987 році.

Суть запропонованих методів полягала в тому, що виготовлення протеза або його частин здійснювалося в автоматичному режимі на фрезерних

верстатах з цифровим програмним керуванням на основі попередньо створеної віртуальної комп'ютерної моделі. Процес моделювання, у свою чергу, відбувався на основі геометрії протезного ложа та даних оклюзійної поверхні протилежних зубів, отриманих у процесі тривимірного сканування поверхні.

На кожному підприємстві працює свої система контролю та обліку, які мають не тільки схожі, но і різноманітні можливості. В силу різноманітних факторів вони не можуть автоматично застосовані на інших підприємствах.

Автоматизована система, яка розроблюється в дані роботі призначена для роботи на новому підприємстві ТОВ «Укрпротез». Автоматизована повинна мати можливість надавати всю різноманітну інформацію керівництву та замовникам. За допомогою системи відбувається контроль термінів, якості виконання замовлення взагалі та окремо на кожному етапі. Забезпечує можливість отримувати для керівництва інформацію про кількість та типи замовлень, про стан оплати та розмір заборгованості, на якому етапі виготовлення знаходиться в даний час замовлення, терміни поставки комплектуючих елементів, внесення змін в термін виготовлення, якщо вони обґрунтовані з форс-мажорними обставинами, облік та відгуки на зауваження та пропозиції.

Після аналізу існуючих автоматизованих систем на підприємствах з протезування, враховуючи особливості технологічного процесу, засобів та методів виготовлення елементів протезів, їхні переваги та недоліки, було прийнято рішення, про необхідність та доцільність розроблення власної системи.

Один із головних аргументів на необхідність розроблення власної системи є те, що вона буде забезпечувати виконання різноманітних функцій, які характерні та необхідні даному підприємству.

Для даної система можна створити програмне забезпечення із урахуванням специфічних вимог і особливостей даного підприємства, що дозволить максимально оптимізувати процеси, забезпечити точність обліку та зручність в роботі при цьому отримати мінімальні витрати. Крім того,

розроблення власної системи дозволяє забезпечити повний контроль над розробкою, впровадженням і підтримкою системи, можна буде впроваджувати нові функції, вносити зміни та адаптувати систему до змін.

Постановка задачі є важливим етапом у процесі розробки будь-якого проекту, включаючи програмні застосунки. Вона визначає мету проекту, його вимоги та обмеження, і створює основу для подальшої роботи. Правильна постановка задачі допомагає забезпечити чітке розуміння вимог та очікувань щодо програмного застосунку, а також визначає основні критерії успіху для подальшої розробки та оцінки результатів. Перш за все, постановка задачі має визначати цілі проекту. Це означає визначення того, що саме має досягнути програмний застосунок. Далі, постановка задачі має чітко визначати вимоги до програмного застосунку. Це можуть бути функціональні вимоги, такі як функції і можливості, які має мати програма, або нефункціональні вимоги, які стосуються продуктивності, безпеки, надійності тощо. Окрім цього, постановка задачі встановлює фундамент для подальшої розробки та допомагає визначити основні критерії успіху. Завданням цієї роботи є розробка автоматизованої системи корекції приймальної гільзи у протезуванні.

Визначимо основні функції, які буде виконувати розроблена система:

- створювати обліковий список елементів різних типів із різними унікальними характеристиками;
- додавати елементи до списку, із інформацію про об'єкт
- редагувати інформацію про обраний елемент списку;
- сортувати елементи списку за різними критеріями;
- відображати інформації у віконному форматі

2 РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ

2.1 Вимоги до нової системи

Автоматизована система керування – сукупність автоматичних керуючих пристроїв, які за участі людини забезпечують функціонування об'єктів керування. АСК здійснює збирання інформації з об'єкта керування, передає, перетворює та опрацьовує її, формує команди управління та виконує їх на керованому об'єкті, тобто здійснює ті функції, які найлегше піддаються автоматизації ДСТУ 2226-93 «Автоматизовані системи. Терміни та визначення»

Автоматизована система керування наприклад підприємств, технологічних процесів, установ, навчальних закладів та інших сферах діяльності людини, це великий комплекс різноманітних як технічних засобів та і програмного забезпечення та баз даних. Дані системи виконують великий спектр різноманітних функцій та складається з багатьох різних, але взаємозалежних підсистем: управління, контроль за виробництвом, контролю параметрів технологічного процесу, як частина загальної системи працює автоматизована система «Склад», фінансова підсистема, правова та інші. Приклад системи виготовлення протезів подана на рисунку 2.1.

Дана система не охоплює весь спектр необхідних функцій, які виконує система зображена на рисунку 2.1. Вона повинна мати можливість працювати самостійно, наприклад як підсистема, склад, бухгалтерія, а також мала би можливість функціонувати з іншими фрагментами повної системи. Основна вимога до нової системи: відслідковування своєчасності виконання замовлення та якості роботи на окремих етапах виготовлення виробів і застосовується автоматизована система. Одним із основних фрагментів системи є база даних, в якій зберігається вся необхідна інформація для забезпечення своєчасного та якісного виготовлення протезів. Проаналізуємо відомі типи БД та виберемо оптимальний тип для даної системи.

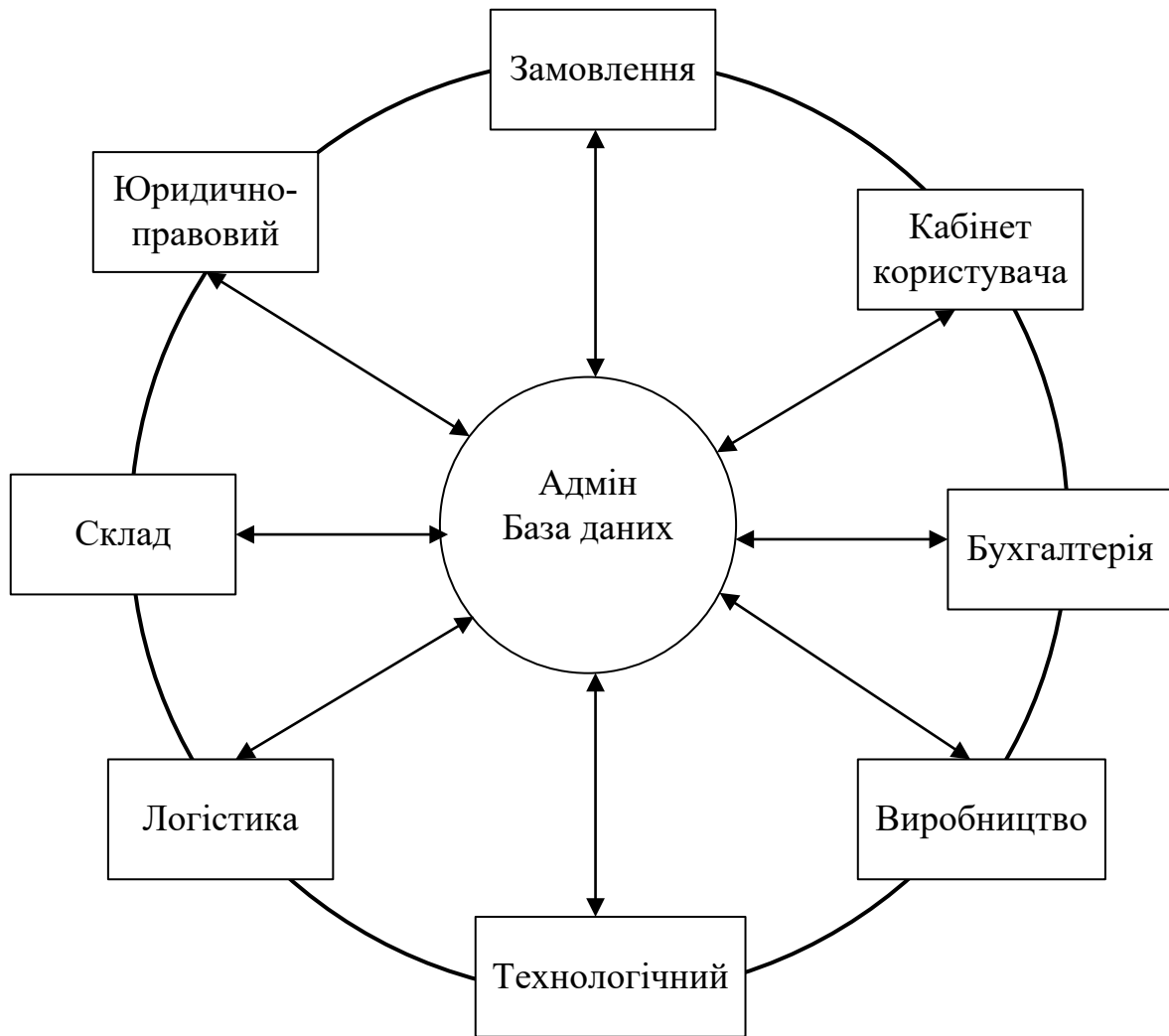


Рисунок 2.1 – Загальна структура автоматизованої системи виробництва протезів

2.2 Розробка бази даних системи

Система управління базами даних (СУБД) – це програмне забезпечення для ефективного, зручного і безпечного зберігання даних у БД, організації пошуку в ній та виведенні даних на вимогу користувачів .

Вибір типу базами даних впливає на функціональні можливості програмного забезпечення, тобто які функції, які операції зможе в подальшому виконувати нове програмне забезпечення.

Проведемо аналіз існуючих баз даних [15], з метою визначення

оптимального варіанту для застосування в даній роботі на підставі технічного завдання.

Найпростіші види баз даних. Найпростіший спосіб зберігання даних - текстові файли. Колись він широко використовувався, але зараз використовується для роботи з невеликими обсягами інформації.

особливості:

- обмежений тип і рівень складності інформації, що зберігається;
- важко встановити зв'язки між компонентами даних;
- відсутність паралелізму функцій;
- практичний лише для систем з невеликими потребами читання та запису;
- використовується для зберігання конфігураційних даних;
- немає необхідності використовувати стороннє програмне забезпечення.

Ієрархічні бази даних. У цьому типі бази даних існують зв'язки між об'єктами. В ієрархічних базах даних створюється деревоподібна структура, у якій записи класифікуються відповідно до їх зв'язку з основним ланцюжком записів.

Особливості:

- інформація організована у вигляді дерева по відношенню «предок-нащадок»;
- кожен запис не може мати більше одного батька;
- зв'язок між записами здійснюється у вигляді фізичних показників;
- відносини «багато до багатьох» неможливі.

Бази які значно збільшують функціональність ієрархічних баз даних називаються мережевими базами, вони мають такі можливості: записи можуть мати більше одного батька. Так можна моделювати складні відносини.

Особливості:

- мережеві бази даних представлені не деревом, а загальним графом
- обмежується тими самими шаблонами доступу, що й ієрархічні бази даних.

Найвідоміші і найстаріші бази даних є реляційні бази, які широко застосовуються для даних загального призначення. В таких базах таблиці використовуються для даних та зв'язки між ними, які організовані за допомогою таблиць. Кожен стовпець у таблиці має назву та тип. Кожен рядок представляє окремий запис або дані в таблиці, які містять значення для кожного стовпця.

Особливості:

- поле в таблиці, яке називається зовнішнім ключем, може містити посилання на стовпці в інших таблицях, щоб їх можна було зв'язати;
- високоорганізована структура та гнучкість роблять реляційні бази даних потужними та адаптованими до різних типів даних;
- для доступу до даних використовується мова структурованих запитів (SQL);
- надійний вибір для багатьох застосувань.

Приклади:

- MySQL
- MariaDB
- PostgreSQL
- SQLite

Нереляційні бази даних. NoSQL — це група типів баз даних, де операції з базою даних організовані інакше, ніж стандартний реляційний шаблон.

Бази даних ключ-значення. У базах даних «ключ-значення» для зберігання інформації необхідно вказати ключ і об'єкт даних для зберігання. Наприклад, JSON-об'єкт, зображення чи текст. Щоб зробити запит, ви надсилаєте ключ і отримуєте blob-об'єкт.

Особливості:

- до сховища інформації існує швидкий доступ при низьких витратах;
- часто зберігає конфігураційні дані та інформацію про стан даних, представлених словниками або хешами;

- немає жорсткого зв'язку між даними, тому такі бази даних часто зберігають різні типи даних одночасно;
- розробник несе відповідальність за визначення схеми іменування ключів і забезпечення відповідного типу/формату значення.

Бази даних або сховища документів, орієнтовані на документи, мають спільну базову семантику доступу та пошуку сховищ ключів і значень. Такі бази даних також використовують ключі для унікальної ідентифікації даних. Різниця між сховищами ключ-значення та базами даних документів полягає в тому, що замість зберігання blob-об'єктів бази даних, орієнтовані на документи, зберігають дані в структурованих форматах – JSON, BSON або XML.

Особливості:

- у базі даних немає окремого формату чи схеми;
- кожен документ може мати свою внутрішню структуру;
- бази даних документів є хорошим вибором для швидкої розробки;
- ви можете будь-коли змінити властивості даних, не змінюючи структуру чи самі дані.

Графові бази даних. Замість встановлення зв'язків із таблицями та зовнішніми ключами бази даних графів встановлюють зв'язки за допомогою вузлів, ребер і властивостей.

Бази даних графів представляють дані як окремі вузли, які можуть мати будь-яку кількість пов'язаних з ними властивостей.

Особливості:

- мережевий зовнішній вигляд;
- зосередьтеся на зв'язках між елементами;
- наочно ілюструє зв'язок між типами даних;
- для переходу між елементами не обов'язково йти крок за кроком;
- немає обмежень щодо видів зв'язку.

Стовпчасті бази даних належать до сімейства баз даних NoSQL, але зовні вони нагадують реляційні бази даних. Як і реляційні бази даних,

стовпчасті бази даних зберігають дані за допомогою рядків і стовпців, але з іншим співвідношенням між елементами.

У реляційних базах даних усі рядки повинні відповідати фіксованій схемі. Схема визначає, які стовпці мають бути в таблиці, типи даних та інші критерії. У колонкових базах даних замість таблиць є структури – «колонка родини». Сімейства містять рядки, кожна з яких визначає власний формат. Рядок складається з унікального ідентифікатора, який використовується для пошуку, після якого йде набір імен стовпців і значень.

Особливості:

- • БД зручні при роботі з додатками, що вимагають високої продуктивності;
- • дані та метадані доступні за допомогою єдиного ідентифікатора;
- • гарантоване розміщення всіх даних із рядка в одному кластері, що спрощує сегментацію та масштабування даних.

Бази даних часових рядів призначені для збору та керування даними, які змінюються з часом [16]. Більшість таких баз даних організовані в структури, які записують значення для одного елемента. Наприклад, ви можете створити таблицю для відстеження температури процесора. Всередині кожне значення складається з мітки часу та показника температури. Таблиця може мати кілька показників.

Особливості:

- • орієнтований на запис;
- • призначений для обробки постійного потоку вхідних даних;
- • продуктивність залежить від кількості відстежуваних елементів, інтервалу опитування між записом нових значень і фактичного завантаження даних.

Комбіновані типи даних включають бази даних NewSQL і мультимодельні бази даних.

Бази даних NewSQL успадковують реляційну структуру та семантику, але побудовані з використанням більш сучасних, масштабованих структур, які

забезпечують більшу масштабованість, ніж реляційні бази даних, і вищі гарантії узгодженості, ніж NoSQL. Компроміс між узгодженістю та доступністю є фундаментальною проблемою для розподілених баз даних, описаних теоремою CAP.

Особливості:

- можливість горизонтального масштабування;
- висока доступність;
- висока продуктивність і тиражування;
- мала функціональність і гнучкість;
- значна витрата ресурсів і необхідність спеціальних знань для роботи з базою даних.

Мультимодельні бази даних - бази даних, що поєднують у собі функціональність кількох типів баз даних. Переваги такого підходу очевидні - одна і та ж система може використовувати різні представлення для різних типів даних.

Загальне розташування даних з кількох типів баз даних в одній системі дозволяє виконувати нові операції, які інакше були б складними або неможливими [17]. Наприклад, багатомодельні бази даних можуть дозволити користувачам отримувати доступ до даних, що зберігаються в різних типах баз даних, і керувати ними в рамках одного запиту, а також підтримувати узгодженість даних при виконанні операцій, які змінюють інформацію одночасно в декількох системах.

Особливості:

- допомога в зниженні навантаження на СУБД;
- можливість розширення до нових моделей у міру зміни потреб без зміни базової інфраструктури;
- забезпечити безперервний доступ до даних і легкий розподіл;
- лінійно масштабована та проста у розробці.

Висновок. Найбільш вдалим, оптимальним варіантом є використання реляційної СУБД.

Самі найпопулярніші СУБД:

- MS SQL;
- MySQL;
- Postgre SQL;
- Oracle.

В дані роботі, з метою забезпечення найвищої ефективності, надійності та масштабованості для реалізації бази даних було обрано MySQL. Дана система керування базами даних здатна оптимально обробляти великі обсяги інформації, забезпечуючи при цьому швидку реакцію на запити та зберігаючи консистентність даних. MySQL, завдяки своїй великій популярності та активній спільноті розробників, є стандартом у сфері реляційних баз даних, що робить його першочерговим вибором для широкого спектру проектів, від невеликих веб-сайтів до великих корпоративних систем .

Виділяються такі популярні клієнти для управління базами даних MySQL, як:

- MySQL Workbench;
- phpMyAdmin;
- Navicat;
- SQLyog;
- HeidiSQL.

У свою чергу, був обраний phpMyAdmin [20], як інструмент для адміністрування бази даних з огляду на його потужний та інтуїтивно зрозумілий веб-інтерфейс. Цей інструмент надає розширені можливості для керування базою даних, включаючи виконання складних запитів, експорт та імпорт даних, а також налаштування прав доступу користувачів. Завдяки phpMyAdmin, адміністраторам не потрібно мати глибоких знань SQL для ефективної роботи з базою даних, що робить його важливим інструментом для будь-якого проекту, що використовує MySQL. Інтеграція phpMyAdmin з MySQL робить процес управління базою даних більш ефективним та зручним, забезпечуючи при цьому високу рівень безпеки та стабільності.

3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Функціонал програми

На сьогоднішній день технології стрімко розвиваються у всіх напрямках повсякденного життя. Не виключенням будуть медичні та інженерні розробки. Одним із ключовим моментом медичної індустрії є постійний збір, аналіз та контроль даних. В деяких моментах ці аспекти можуть коштувати життя людині. Тому розроблена мною програма повинна покращити процес збору та аналізу даних в медичній сфері.

Завдяки інтуїтивно зрозумілому інтерфейсу, моя програма дозволяє користувачам легко отримувати доступ до даних про клієнтів, етапи розробки протезу та заміри протезу. Функціонал пошуку та фільтрації даних дозволяє швидко знаходити необхідну інформацію, а можливість додавання власних даних розширює можливості програми та дозволяє враховувати індивідуальні потреби кожного клієнта.

Створена програма не лише спрощує процес управління даними, а й сприяє покращенню якості надання послуг у галузі медичних та інженерних технологій. Забезпечуючи доступність, точність та зручність у використанні, вона стає невід'ємним інструментом для фахівців, що займаються розробкою та виготовленням протезів для клієнтів.

В подальшому розвитку даної програми користувач зможуть додавати і редагувати записи в базі даних.

Основний функціонал програми:

Керування даними: користувачі програми не матимуть проблем з керуванням великим обсягом даних, який знаходиться в базі даних. Дані про клієнтів, етапи розробки протезу та заміри протезу – усі ці дані будуть представлені користувачеві в простій та зрозумілій формі. Крім того, користувач ніколи не заплутається у всіх цих даних через їх логічне представлення через Windows Forms, яке надає середовище Visual Studio.

Пошук та фільтрація даних: неможливо уявити керування великим обсягом даних без пошуку по ним. Моя програма надає універсальний пошук, завдяки якому потенційний користувач легко знаходитиме необхідну для нього інформацію, яка знаходиться в таблицях бази даних. Пошук і фільтрація моєї програми орієнтовані на різні способи пошуку: за ім'ям та прізвищем клієнта, датою укладення договору, статтю, віком, вагою та багатьма іншими характеристиками всередині таблиць бази даних. Цей функціонал надає більше можливостей з пошуку та фільтрації даних, що полегшує користувачу моніторинг та аналіз даних.

Збереження інформації: в подальшому розвитку програми збереження буде ключовим фактором доповнення бази даних. Усі дані безпечно додаватимуться до бази даних на серверах програми. Їх цілісність та захищеність забезпечуватимуться внутрішніми протоколами захисту та збереження, адже такі дані є конфіденційними.

Додавання власних даних: додавання даних до бази даних також є важливим функціоналом, який буде доданий в майбутньому. Оновлення даних є важливим етапом в роботі з базою даних.

Зручний інтерфейс: метод роботи з Windows Forms у Visual Studio є зручним та ефективним через свою інтуїтивну зрозумілість та комфортність в користуванні. У користувачів не виникатимуть проблеми з швидкою та ефективною роботою з даною програмою. Зручне представлення даних є важливим фактором при роботі з даною програмою.

3.2 Вибір мови програмування

Для створення графічного інтерфейсу використовувалася мова програмування C# [18] у середовищі Windows Forms. Windows Forms є фреймворком, спеціально призначеним для розробки десктопних додатків в середовищі Microsoft. З ним розробники мають можливість легко створювати інтерфейси для Windows додатків із використанням різноманітних елементів

управління.

Однією з ключових переваг Windows Forms є його інтеграція з середовищем розробки Visual Studio, що надає зручність та продуктивність у процесі створення додатків. Розробники можуть швидко перетягувати та розміщувати елементи управління на формі, налаштовувати їх властивості та забезпечувати взаємодію з логікою програми.

C#, як мова програмування, володіє сучасним синтаксисом та багатофункціональністю, що робить її ефективною для розробки десктопних додатків. Вона підтримує об'єктно-орієнтований підхід, а також має широкий спектр бібліотек і класів, які полегшують роботу з різними аспектами програмування.

За допомогою Windows Forms у поєднанні з C#, розробники можуть ефективно створювати інтерфейси з усіма необхідними елементами, такими як кнопки, тексти, списки та інші, для створення зручного та користувацькиорієнтованого досвіду взаємодії з десктопними додатками на платформі Windows.

3.3 Вибір редактора коду та його особливості

Для написання коду та подальшої реалізації програми було вибрано найзручніший як на мою думку редактор коду

Visual Studio 2022. Visual Studio [19] - це потужний засіб розробника, який можна використовувати для виконання всього циклу розробки в одному місці. Це комплексне інтегроване середовище розробки (IDE), яке можна використовувати для запису, редагування, налагодження та складання коду, а потім розгортання програми. Крім редагування і налагодження коду Visual Studio включає компілятори, засоби завершення коду, управління версіями, розширення і багато іншого, щоб поліпшити кожен етап процесу розробки програмного забезпечення.

Переваги редактора Visual Studio:

Широкий спектр підтримуваних мов програмування: Visual Studio 2022 підтримує багато мов програмування, включаючи C#, C++, F#, Visual Basic, JavaScript, Python та інші. Це робить його універсальним інструментом для розробки програмного забезпечення на різних платформах і для різних цілей.

Багатофункціональність: Visual Studio 2022 має широкий набір інструментів, включаючи інтегровану систему керування версіями, редагування коду з підсвічуванням синтаксису, автодоповненням, відлагодженням, підтримкою юніт-тестування, інтегрованою системою збірки та багато іншого.

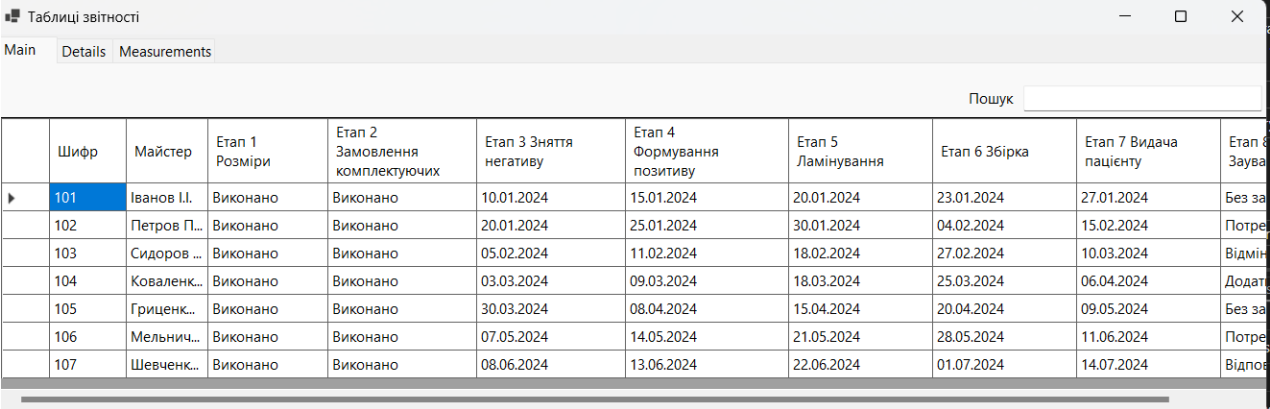
Розширюваність: Visual Studio 2022 дозволяє розробникам встановлювати різні додатки та розширення для покращення функціональності IDE. Це дозволяє налаштувати середовище розробки під конкретні потреби кожного розробника.

Підтримка розробки під різні платформи: Visual Studio 2022 дозволяє розробляти програмне забезпечення для різних платформ, таких як Windows, Android, iOS, Linux, Web і багато інших. Це робить його універсальним інструментом для розробки кросплатформених додатків.

Загалом, Visual Studio 2022 є потужним та універсальним інструментом для розробки програмного забезпечення з багатьма функціями та можливостями, які допомагають розробникам.

3.4 Результати роботи програмного забезпечення

Нова розроблена програма складається з трьох підпрограм: перша головна (рис. 3.1) застосовується для аналізу і відслідковування термінів виконання замовлення, а також має можливість надавати і іншу інформацію.



Шифр	Майстер	Етап 1 Розміри	Етап 2 Замовлення комплектуючих	Етап 3 Зняття негативу	Етап 4 Формування позитиву	Етап 5 Ламінування	Етап 6 Збірка	Етап 7 Видача пацієнту	Етап 8 Заува
101	Іванов І.І.	Виконано	Виконано	10.01.2024	15.01.2024	20.01.2024	23.01.2024	27.01.2024	Без за
102	Петров П...	Виконано	Виконано	20.01.2024	25.01.2024	30.01.2024	04.02.2024	15.02.2024	Потре
103	Сидоров ...	Виконано	Виконано	05.02.2024	11.02.2024	18.02.2024	27.02.2024	10.03.2024	Відмін
104	Коваленк...	Виконано	Виконано	03.03.2024	09.03.2024	18.03.2024	25.03.2024	06.04.2024	Додат
105	Гриценк...	Виконано	Виконано	30.03.2024	08.04.2024	15.04.2024	20.04.2024	09.05.2024	Без за
106	Мельнич...	Виконано	Виконано	07.05.2024	14.05.2024	21.05.2024	28.05.2024	11.06.2024	Потре
107	Шевченк...	Виконано	Виконано	08.06.2024	13.06.2024	22.06.2024	01.07.2024	14.07.2024	Відпо

Рисунок 3.1 – Відображення таблиці основної програми

На рисунках 3.2-3.4 представлено результат вибору за різним параметрам, за шифру замовлення, за датою та за ключовим словом.

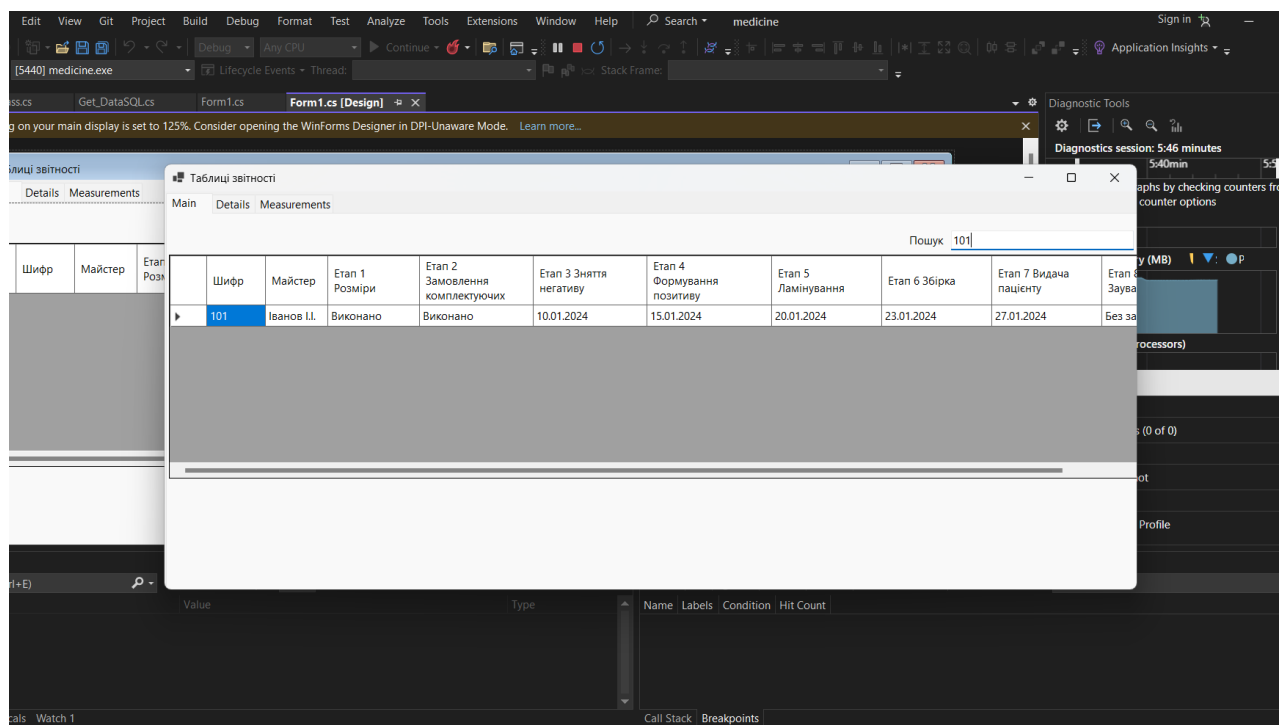


Рисунок 3.2 – Результати пошуку за ознакою «Шифр»

В основному вікні користувач може здійснювати пошук за шифром договору. Це надає швидкий та зручний доступ при пошуку необхідної інформації. Кожен договір має індивідуальний шифр.

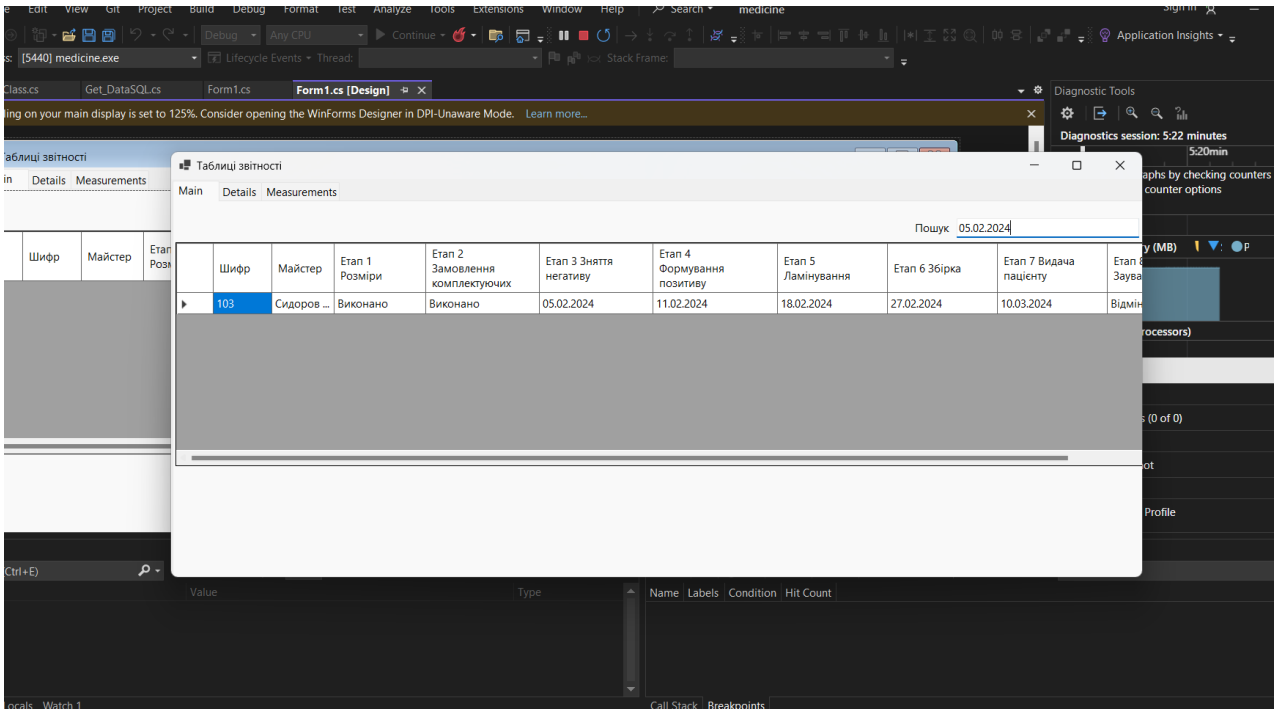


Рисунок 3.3 – Результати пошуку за ознакою «Дата»

Також користувач може здійснювати пошук за датами, що є корисним в контексті часових рамок.

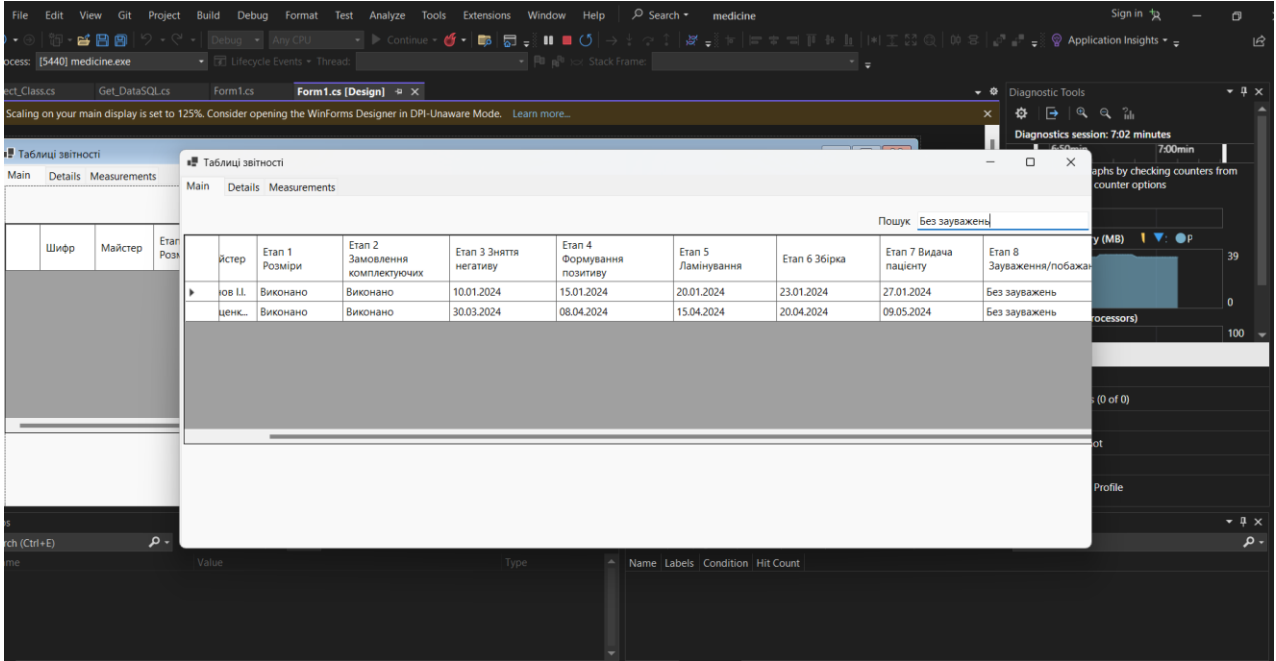


Рисунок 3.4 – Результати пошуку за ключовими словами

Пошук за ключовими словами також важливий. В залежності від

потреби та критеріїв пошуку користувач може здійснювати швидкий аналіз всіх таблиць в даній базі даних.

На рисунку 3.5 зображена початкова таблиця другої підпрограми, яка надає детальну інформацію про замовлення.

Таблиці звітності

MainDetailsMeasurements

Пошук

	Дата подання бланку	ПІБ	Шифр	Рівень мобільності (1-2-3-4)	Стать	Вік	Зріст	Вага	Нога	Вражена кінцівка	Майстер	Термін виконання	Ціна
►	01.01.2024	Ковален...	101	3	Ч	45	180 см	75 кг	ліва	гомілка	Іванов І.І.	28.01.2024	5000
	15.01.2024	Петренк...	102	2	Ж	50	165 см	60 кг	права	стегно	Петров П.П.	10.02.2024	6000
	01.02.2024	Сидорен...	103	4	Ч	30	175 см	70 кг	ліва	гомілка	Сидоров С.С.	10.03.2024	7000
	28.02.2024	Іваненк...	104	1	Ж	40	170 см	65 кг	права	стегно	Коваленко К.К.	05.04.2024	5500
	25.03.2024	Гриценк...	105	5	Ч	35	182 см	80 кг	ліва	гомілка	Гриченко Г.Г.	10.05.2024	4800
	01.05.2024	Мельни...	106	3	Ч	55	178 см	85 кг	права	стегно	Мельниченко ...	16.06.2024	6200
	01.06.2024	Шевчен...	107	4	Ж	60	160 см	70 кг	ліва	гомілка	Шевченко Ш.Ш.	15.07.2024	7000

Рисунок 3.5 – Відображення таблиці основної програми «Details»

На рисунка 3.6-3.8 показано результати роботи програми по пошуку інформації за різними параметрами, а саме, за датою (рис 3.6), за статтю замовника (рис. 3.7), за розміром заборгованості (рис. 3.8).

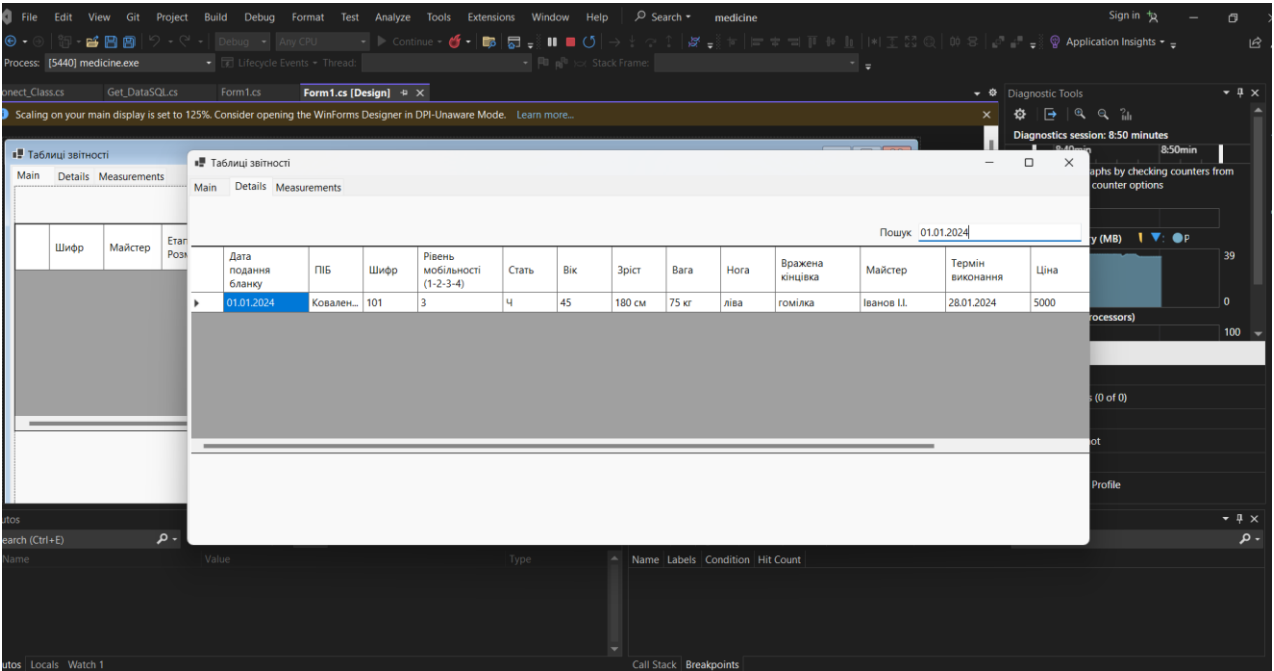


Рисунок 3.6 – Результати пошуку за датою

В допоміжних таблицях користувач також може здійснювати пошук за датами, як і в основній таблиці. Доступність універсального пошуку спрощує аналіз шуканих даних.

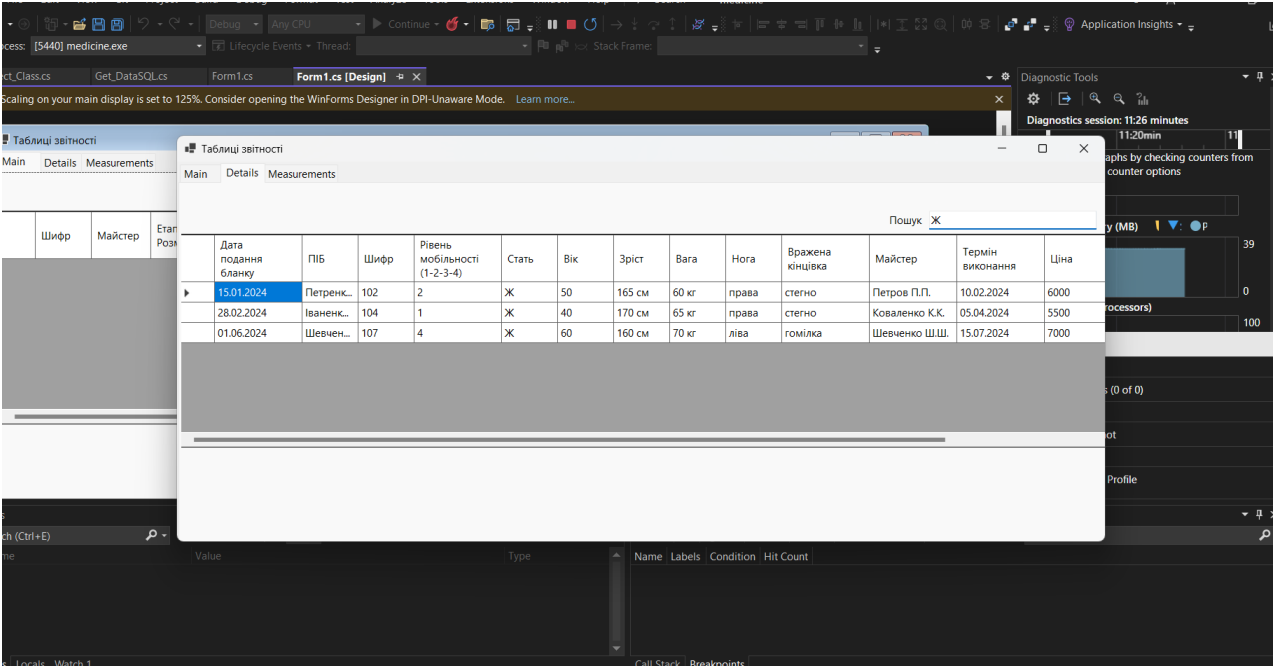


Рисунок 3.7 – Результати пошуку за статтю

Також користувачу доступний пошук за статтю, що теж спрощує аналіз даних в таблицях.

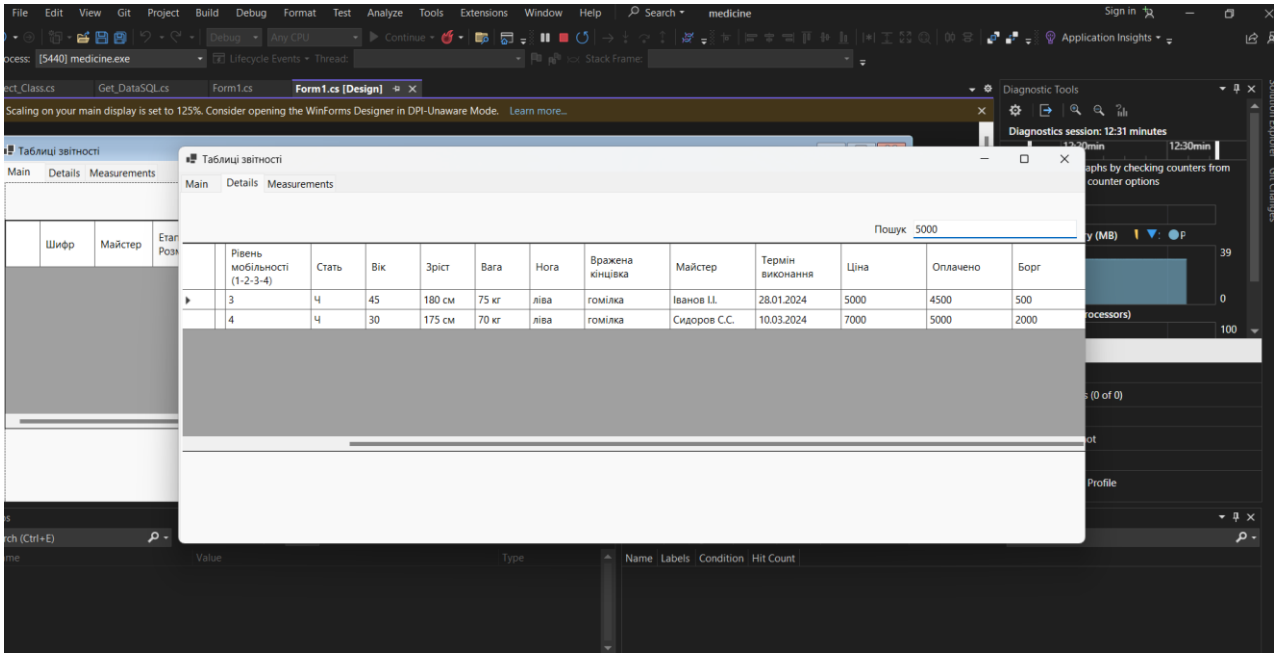


Рисунок 3.8 – Результати пошуку за ціною, оплаченим рахунком і боргом

Важливим є пошук за ціною, оплаченим рахунком і боргом. Це надає корисну інформацію користувачеві.

На рисунку 3.9 зображена таблиці третьої програми, яка відображає антропологічні дані замовника. Також має можливість надавати інформацію за різним параметрам, шифр (рис. 3.9), тип ампутації (рис. 3.10)

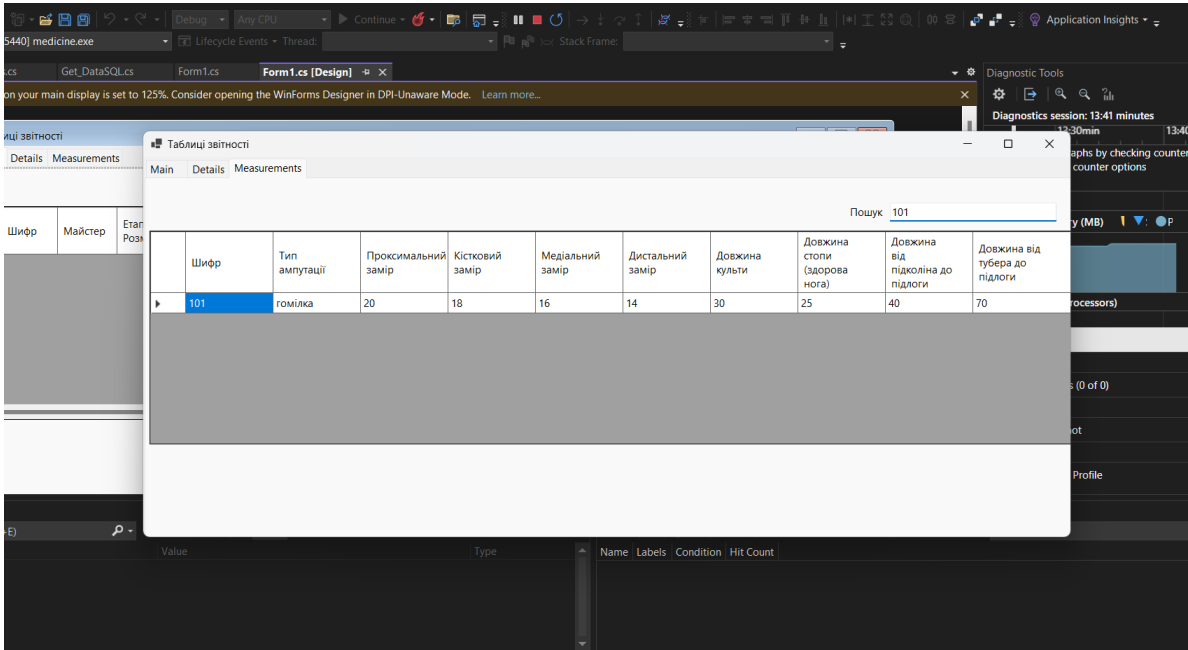


Рисунок 3.9 – Результати пошуку за шифром

Пошук за шифром в таблиці заміри є критично важливим для швидкого аналізу та роботи з інформацією щодо замірів протеза.

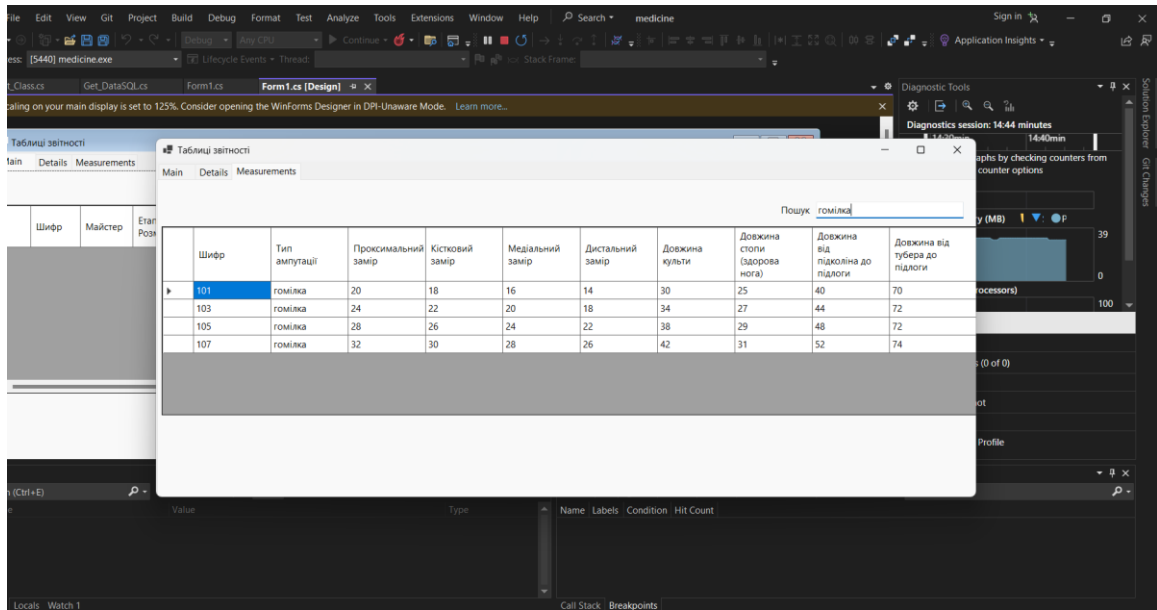


Рисунок 3.10 – Результати пошуку за ключовими словами

ВИСНОВКИ

У даній бакалаврській дипломній роботі було проведено аналіз автоматизованої системи виробництва протезів, створений програмне забезпечення для покращеного контролю над базою даних з відомостями про етапи вироблення протезів. Завдяки додатку користувачі зможуть швидко знаходити необхідну інформацію та взаємодіяти з нею.

Проаналізувавши поставлену задачу визначили та реалізували архітектуру програми. Склали ERM-діаграму програмної бази даних. Проаналізували сучасні СУБД та обрали phpMyAdmin як найбільш доступну, просту та зрозумілу СУБД. Також проаналізували сучасні інструменти для реалізації програми і зробили вибір у бік мови програмування C# та середовища Visual Studio 2022 (Windows Forms), як зрозумілих та структурованих інструментів для вирішення поставленого завдання.

Після створення програми та налагодження зв'язку між базою даних в СУБД phpMyAdmin та програмним середовищем у Visual Studio 2022, провели тестування функціоналу програми та виправили помилки пов'язані з неправильною архітектурою та логікою бази даних та невідповідностями в методах, прописаних в коді програми.

Результатам даної бакалаврської дипломної роботи є розроблена структурована програма перегляду та пошуку даних в базі даних про розробку вставних протезів. Дана програма надає зручний інтерфейс та можливість швидко знаходити необхідну інформацію по всій базі даних.

ПЕРЕЛІК ПОСИЛАНЬ

1. Конструювання та технології виготовлення протезів нижніх кінцівок: навч. посіб. / А. Д. Салєєва, О. Г. Аврунін, П. О. Басєв, С. В. Корнєєв, Я. В. Носова, І. В. Кабаненко, М. В. Зайцев, Т. О. Трофименко, І. Л. Тимофєєв. - Харків: ХНУРЕ, 2023. - 481 с.
2. Протезування та ортезування: курс лекцій дистанційного навчання. Модуль І. Протезування нижніх кінцівок. – Нюрберґ, Німеччина (ISPO / HumanStudy.V. / DonBoscoUniversity), 2010.
3. Гільза для протезів стегна: патент на винахід України № 59759 / І.Л. Тимофєєв та ін.; заявник УкрНДІпротезування; опубл. 15.10.2002. Бюл. № 10.
4. Складання та регулювання протезів нижніх кінцівок: методичні рекомендації. – Харків: УкрНДІпротезування, 2013. – 50 с.
5. Призначення конструкцій гільз до протезів нижніх кінцівок: навчально-методичний посібник. – Харків: УкрНДІпротезування, 2009. – 50 с
6. Етапи 3D-друку URL: <https://3dway.com.ua/blog/stages-of-3d-printing> (дата звернення: 23.05.24).
7. Гільзи та системи кріплення протезів нижніх кінцівок URL https://langs.physio-pedia.com/uk/lower-limb-prosthetic-sockets-and-suspension-systems-uk/#cite_note-:8-2 (дата звернення: 23.05.24).
8. Режим доступу 3D друк DIY протезів у 2021 році <https://3dreams.com.ua/ua/3d-%D0%B4%D1%80%D1%83%D0%BA-%D0%BF%D1%80%D0%BE%D1%82%D0%B5%D0%B7%D1%96%D0%B2-%D1%83-2021-%D1%80%D0%BE%D1%86%D1%96/> (дата звернення: 23.05.24).
9. Ціж Л. М. Актуальні проблеми забезпечення технічними засобами реабілітації осіб з інвалідністю в Україні / Л. М. Ціж. // Вісник Запорізького національного університету: Збірник наук. статей. - Фізичне виховання та спорт. – 2017. – №1. – С. 183–191.

10. Травматологія і ортопедія: підручник для студ. вищих мед. навч. закладів / за ред. Г.Г. Голки, О.А.Бур`янова, В.Г.Климовицького. – 126 Вінниця: Нова книга, 2014. – 416с.
11. Гур'єв С.О., Лисун Д.М., Кушнір В.А., Садик С.П., Кураченко І.П.Ампутації нижніх кінцівок унаслідок сучасних бойових дій (клініко-анатомічний аспект). Травма. Том 19, № 4, 2018, с.5-8
12. Лябах А.Б. Ампутації нижньої кінцівки.-К.:Стилос, 2022.-186с.
13. Протезування та ортезування у фізичній терапії: навчальний посібник /Я.Ф. Філак, Ф.Г. Філак – Ужгород: ФОП Сабов А.М. – 2018. – 102 с.
14. Що таке C# [Електронний ресурс]: Режим доступу: <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/overview> – Назва з екрану
15. Савчук, Т. О. Організація баз даних та знань [Текст]: лабораторний практикум. Ч. 2 / Т. О. Савчук ; ВНТУ. – Вінниця : ВНТУ, 2016.
16. Руденко, В. Д. Бази даних в інформаційних системах [Текст]: навчальний посібник / В. Д. Руденко ; за ред. В. Ю. Бикова. – К. : Фенікс, 2010. – 240 с. – ISBN 978-966-651-809-8
17. Петух, А. М. Бази даних. Мови запитів, управління транзакціями, розподілена обробка даних [Електронний ресурс] / А. М. Петух, О. В. Романюк, О. Н. Романюк. – Електрон. текст. дані. – Вінниця : ВНТУ, 2016.
18. Що таке Visual Studio [Електронний ресурс]: Режим доступу <https://learn.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs2022> – Назва з екрану
19. Як керувати базами даних за допомогою phpMyAdmin :<https://www.namecheap.com/support/knowledgebase/article.aspx/9540/2180/how-to-manage-databases-with-phpmyadmin/>
20. Ласкаво просимо до документації phpMyAdmin! :<https://docs.phpmyadmin.net/en/latest/>

ДОДАТКИ

ДОДАТОК А

**ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ**

Назва роботи: « Автоматизована система корекції приймальної гільзи у протезуванні»

Тип роботи: Бакалаврська дипломна робота
(БДР, МКР)

Підрозділ КСУ, ФІТА
(кафедра, факультет)

Показники звіту подібності Unicheck

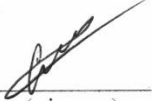
Оригінальність 87,7% Схожість 12,3%

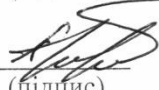
Аналіз звіту подібності (відмітити потрібне)

- ☐ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку  Володимир ДУБОВОЙ
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи  Віталій СТУКЕЛЬМАН
(підпис) (прізвище, ініціали)

Керівник роботи  Василь ПРИСЯЖНЮК
(підпис) (прізвище, ініціали)

Додаток Б
(обов'язковий)
ВНТУ

ЗАТВЕРДЖЕНО
Зав. кафедри КСУ ВНТУ,
д.т.н., проф.

 **В'ячеслав КОВТУН**

“ 24 ” травня 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання бакалаврської дипломної роботи

Автоматизована система корекції приймальної гільзи у протезуванні

Студент групи

 **АКІТ-22мс Віталій СТУКЕЛЬМАН**

Керівник

 **ст.викл.каф. КСУ, Василь ПРИСЯЖНЮК**

1. Назва та галузь застосування

1.1. Назва – Автоматизована система корекції приймальної гільзи у протезуванні.

1.2. Галузь застосування – Виробництво

2. Підстава для проведення розробки.

Тема бакалаврської дипломної роботи затверджена наказом по ВНТУ № 80 від 11 березня .2024 р.

3. Мета та призначення розробки.

Метою бакалаврської дипломної роботи підвищення ефективність процесів контролю та обліку виробництва протезів, шляхом розробки програмного забезпечення

4. Джерела розробки.

Бакалаврська дипломна робота виконується вперше. В ході проведення розробки повинні використовуватись такі документи:

1. Конструювання та технології виготовлення протезів нижніх кінцівок: навч. посіб. / А. Д. Салєєва, О. Г. Аврунін, П. О. Баєв, С. В. Корнєєв, Я. В. Носова, І. В. Кабаненко, М. В. Зайцев, Т. О. Трофименко, І. Л. Тимофєєв. - Харків: ХНУРЕ, 2023. - 481 с

2. Гільзи та системи кріплення протезів нижніх кінцівок URL https://langs.physio-pedia.com/uk/lower-limb-prosthetic-sockets-and-suspension-systems-uk/#cite_note-:8-2 (дата звернення: 23.05.24)

3. В.М. Дубовой. Методичні вказівки до виконання бакалаврськихбакалавирських робіт для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології». Вінниця :ВНТУ,2022.URL <https://iq.vntu.edu.ua/repository/card.php?id=5929&lang=uk>

5. Вимоги до розробки.

5.1. Перелік головних функцій:

- створювати обліковий список елементів різних типів із різними унікальними характеристиками;

- редагувати інформацію про обраний елемент списку;
- сортувати елементи списку за різними критеріями;
- відображати інформації у віконному форматі.

5.2. Основні технічні вимоги до розробки.

5.2.1. Вимоги до програмної платформи:

- сервер OpenServer

5.2.2. Умови експлуатації системи:

- можливість цілодобового функціонування системи;
- дані оновлюються і є актуальними.
-

6. Стадії та етапи розробки.

6.1 Пояснювальна записка:

1. Аналіз методів, принципів, підходів і засобів реалізації задачі автоматизації процесами в об'єкті управління відповідно до теми дипломної роботи. Постановка задач дослідження «15» травня 2024 р.

2. Визначення технічних характеристик системи «19» травня 2024 р.

3. Розробка програмного забезпечення системи «30» травня 2024 р.

6.2 Графічні матеріали:

1. Розробка моделі системи «24» травня 2024 р.

2. Тестування програмного забезпечення «10» червня 2024 р.

7. Порядок контролю і приймання.

7.1. Хід виконання роботи контролюється керівником роботи. Рубіжний контроль провести до «4» червня 2024 р.

7.2. Атестація проєкту здійснюється на попередньому захисті. Попередній захист бакалаврської дипломної роботи провести до «10» червня 2024 р.

7.3. Підсумкове рішення щодо оцінки якості виконання роботи приймається на засіданні ЕК. Захист бакалаврської дипломної роботи провести до «20» червня 2024 р.

Додаток Б
(обов'язковий)
ВНТУ

ЗАТВЕРДЖЕНО
Зав. кафедри КСУ ВНТУ,
д.т.н., проф.

 **В'ячеслав КОВТУН**

“ 24 ” травня 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

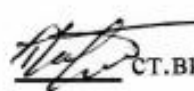
на виконання бакалаврської дипломної роботи

Автоматизована система корекції приймальної гільзи у протезуванні

Студент групи

 **АКІТ-22мс Віталій СТУКЕЛЬМАН**

Керівник

 **ст.викл.каф. КСУ, Василь ПРИСЯЖНЮК**

Вінниця 2024



Рисунок В.1 – Будова протезу стегна



Рисунок В.2 – Куксоприймальна гільза

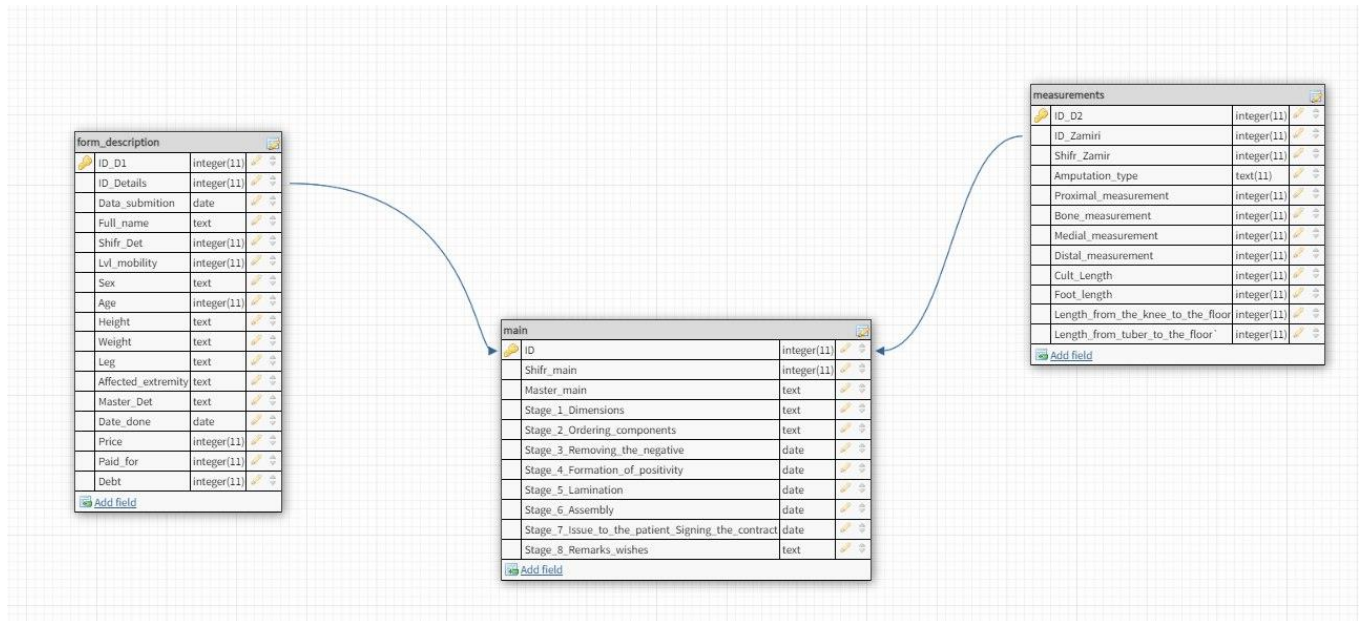


Рисунок В.3 – ERM-діаграма бази даних.

Нова		Містить слово:	
information_schema			
medicine			
Нова			
form_description			
main			
measurements			
mysql			
performance_schema			

Таблиця	Дія	Рядки	Тип	Зіставлення	Розмір	Фрагментовані
☐ form_description	★ Переглянути Структура Пошук Вставити Очистити Знищити	7	InnoDB	utf8mb4_general_ci	32.0 КБ	-
☐ main	★ Переглянути Структура Пошук Вставити Очистити Знищити	7	InnoDB	utf8mb4_general_ci	32.0 КБ	-
☐ measurements	★ Переглянути Структура Пошук Вставити Очистити Знищити	7	InnoDB	utf8mb4_general_ci	32.0 КБ	-
3 таблиці	Всього	21	InnoDB	utf8mb4_general_ci	96.0 КБ	0 Б

Рисунок В.4 – Структура бази даних в середовищі phpMyAdmin.

Сервер: 127.0.0.1База даних: medicineТаблиця: form_description

ПереглянутиСтруктураSQLПошукВставитиЕкспортІмпортПривілеїОпераціїВідстеження

Структура таблиціВид відносин

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
☐	1 ID_D1	int(11)			Ні	Немає		AUTO_INCREMENT	Змінити Знищити Більше
☐	2 ID_Details	int(11)			Так	NULL			Змінити Знищити Більше
☐	3 Data_submission	date			Так	NULL			Змінити Знищити Більше
☐	4 Full_name	text	utf8mb4_general_ci		Так	NULL			Змінити Знищити Більше
☐	5 Shifr_Det	int(11)			Так	NULL			Змінити Знищити Більше
☐	6 Lvl_mobility	int(11)			Так	NULL			Змінити Знищити Більше
☐	7 Sex	text	utf8mb4_general_ci		Так	NULL			Змінити Знищити Більше
☐	8 Age	int(11)			Так	NULL			Змінити Знищити Більше
☐	9 Height	text	utf8mb4_general_ci		Так	NULL			Змінити Знищити Більше
☐	10 Weight	text	utf8mb4_general_ci		Так	NULL			Змінити Знищити Більше
☐	11 Leg	text	utf8mb4_general_ci		Так	NULL			Змінити Знищити Більше
☐	12 Affected_extremity	text	utf8mb4_general_ci		Так	NULL			Змінити Знищити Більше
☐	13 Master_Det	text	utf8mb4_general_ci		Так	NULL			Змінити Знищити Більше
☐	14 Date_done	date			Так	NULL			Змінити Знищити Більше
☐	15 Price	int(11)			Так	NULL			Змінити Знищити Більше
☐	16 Paid_for	int(11)			Так	NULL			Змінити Знищити Більше
☐	17 Debt	int(11)			Так	NULL			Змінити Знищити Більше

Рисунок В.5 – Структура таблиці “form_description”.

Таблиця: form_description

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
☐	1 ID_D1	int(11)			Ні	Немає		AUTO_INCREMENT	Змінити Знищити Більше
☐	2 ID_Details	int(11)			Так	NULL			Змінити Знищити Більше
☐	3 Data_submission	date			Так	NULL			Змінити Знищити Більше
☐	4 Full_name	text	utf8mb4_general_ci		Так	NULL			Змінити Знищити Більше
☐	5 Shifr_Det	int(11)			Так	NULL			Змінити Знищити Більше
☐	6 Lvl_mobility	int(11)			Так	NULL			Змінити Знищити Більше
☐	7 Sex	text	utf8mb4_general_ci		Так	NULL			Змінити Знищити Більше
☐	8 Age	int(11)			Так	NULL			Змінити Знищити Більше
☐	9 Height	text	utf8mb4_general_ci		Так	NULL			Змінити Знищити Більше
☐	10 Weight	text	utf8mb4_general_ci		Так	NULL			Змінити Знищити Більше
☐	11 Leg	text	utf8mb4_general_ci		Так	NULL			Змінити Знищити Більше
☐	12 Affected_extremity	text	utf8mb4_general_ci		Так	NULL			Змінити Знищити Більше
☐	13 Master_Det	text	utf8mb4_general_ci		Так	NULL			Змінити Знищити Більше
☐	14 Date_done	date			Так	NULL			Змінити Знищити Більше
☐	15 Price	int(11)			Так	NULL			Змінити Знищити Більше
☐	16 Paid_for	int(11)			Так	NULL			Змінити Знищити Більше
☐	17 Debt	int(11)			Так	NULL			Змінити Знищити Більше

Рисунок В.6 – Вигляд заповненої даними таблиці “form_description”.

Сервер: 127.0.0.1База даних: medicineТаблиця: main

ПереглянутиСтруктураSQLПошукВставитиЕкспортІмпортПривілеїОпераціїВідстеженняТригери

Структура таблиціВид відносин

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
☐	1 ID	int(11)			Ні	Немає		AUTO_INCREMENT	Змінити Знищити Більше
☐	2 Shifr_main	int(11)			Так	NULL			Змінити Знищити Більше
☐	3 Master_main	text	utf8mb4_general_ci		Так	NULL			Змінити Знищити Більше
☐	4 Stage_1_Dimensions	text	utf8mb4_general_ci		Так	NULL			Змінити Знищити Більше
☐	5 Stage_2_Ordering_components	text	utf8mb4_general_ci		Так	NULL			Змінити Знищити Більше
☐	6 Stage_3_Removing_the_negative	date			Так	NULL			Змінити Знищити Більше
☐	7 Stage_4_Formation_of_positivity	date			Так	NULL			Змінити Знищити Більше
☐	8 Stage_5_Lamination	date			Так	NULL			Змінити Знищити Більше
☐	9 Stage_6_Assembly	date			Так	NULL			Змінити Знищити Більше
☐	10 Stage_7_Issue_to_the_patient_Signing_the_contract	date			Так	NULL			Змінити Знищити Більше
☐	11 Stage_8_Remarks_wishes	text	utf8mb4_general_ci		Так	NULL			Змінити Знищити Більше

Рисунок В.7 – Структура таблиці “main”.

ID	Shifr_main	Master_main	Stage_1_Dimensions	Stage_2_Ordering_components	Stage_3_Removing_the_negative	Stage_4_Formation_of_positivity	Stage_5_Lamination	Stage_6_Assembly	Stage_7_Issue_to_the_patient_Signing_the_o
1	101	Іванов І.І.	Виконано	Виконано	2024-01-20	2024-01-25	2024-01-30	2024-02-04	2024-02-15
2	102	Петров П.П.	Виконано	Виконано	2024-02-05	2024-02-11	2024-02-18	2024-02-27	2024-03-10
3	103	Сидоров С.С.	Виконано	Виконано	2024-03-03	2024-03-09	2024-03-18	2024-03-25	2024-04-06
4	104	Коваленко К.К.	Виконано	Виконано	2024-03-07	2024-04-08	2024-04-15	2024-04-20	2024-05-09
5	105	Гриценко Г.Г.	Виконано	Виконано	2024-05-07	2024-05-14	2024-05-21	2024-05-28	2024-06-11
6	106	Мельниченко М.М.	Виконано	Виконано	2024-06-08	2024-06-13	2024-06-22	2024-07-01	2024-07-14
7	107	Шевченко Ш.Ш.	Виконано	Виконано					

Рисунок В.8 – Вигляд заповненої даними таблиці “main”.

#	Im'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	ID_D2	int(11)			Ні	Немає		AUTO_INCREMENT	Змінити Знищити Більше
2	ID_Zamiri	int(11)			Так	NULL			Змінити Знищити Більше
3	Shifr_Zamir	int(11)			Так	NULL			Змінити Знищити Більше
4	Amputation_type	text	utf8mb4_general_ci		Так	NULL			Змінити Знищити Більше
5	Proximal_measurement	int(11)			Так	NULL			Змінити Знищити Більше
6	Bone_measurement	int(11)			Так	NULL			Змінити Знищити Більше
7	Medial_measurement	int(11)			Так	NULL			Змінити Знищити Більше
8	Distal_measurement	int(11)			Так	NULL			Змінити Знищити Більше
9	Cult_Length	int(11)			Так	NULL			Змінити Знищити Більше
10	Foot_length	int(11)			Так	NULL			Змінити Знищити Більше
11	Length_from_the_knee_to_the_floor	int(11)			Так	NULL			Змінити Знищити Більше
12	Length_from_tuber_to_the_floor	int(11)			Так	NULL			Змінити Знищити Більше

Рисунок В.9 – Структура таблиці “measurements”.

ID_D2	ID_Zamiri	Shifr_Zamir	Amputation_type	Proximal_measurement	Bone_measurement	Medial_measurement	Distal_measurement	Cult_Length	Foot_Length	Length_from_the_knee_to_the_floor	Length_from_tuber_to_the_floor
1	1	101	гомілка	20	18	16	14	30	25	40	70
2	2	102	стєно	22	20	18	16	32	26	42	78
3	3	103	гомілка	24	22	20	18	34	27	44	72
4	4	104	стєно	26	24	22	20	36	28	46	70
5	5	105	гомілка	28	26	24	22	38	29	48	72
6	6	106	стєно	30	28	26	24	40	30	50	71
7	7	107	гомілка	32	30	28	26	42	31	52	74

Рисунок В.10 – Вигляд заповненої даними таблиці “measurements”.

Таблиці звітності

Main Details Measurements

Пошук

	Шифр	Майстер	Етап 1 Розміри	Етап 2 Замовлення комплектуючих	Етап 3 Зняття негативу	Етап 4 Формування позитиву	Етап 5 Ламінування	Етап 6 Збірка	Етап 7 Видача пацієнту	Етап 8 Заува

Рисунок В.11 – Вигляд вікна “Таблиці звітності” створеного додатку.

Таблиці звітності

Main Details Measurements

Пошук

	Шифр	Майстер	Етап 1 Розміри	Етап 2 Замовлення комплектуючих	Етап 3 Зняття негативу	Етап 4 Формування позитиву	Етап 5 Ламінування	Етап 6 Збірка	Етап 7 Видача пацієнту	Етап 8 Заува
▶	101	Іванов І.І.	Виконано	Виконано	10.01.2024	15.01.2024	20.01.2024	23.01.2024	27.01.2024	Без за
	102	Петров П...	Виконано	Виконано	20.01.2024	25.01.2024	30.01.2024	04.02.2024	15.02.2024	Потре
	103	Сидоров ...	Виконано	Виконано	05.02.2024	11.02.2024	18.02.2024	27.02.2024	10.03.2024	Відмін
	104	Коваленк...	Виконано	Виконано	03.03.2024	09.03.2024	18.03.2024	25.03.2024	06.04.2024	Додат
	105	Гриценк...	Виконано	Виконано	30.03.2024	08.04.2024	15.04.2024	20.04.2024	09.05.2024	Без за
	106	Мельнич...	Виконано	Виконано	07.05.2024	14.05.2024	21.05.2024	28.05.2024	11.06.2024	Потре
	107	Шевченк...	Виконано	Виконано	08.06.2024	13.06.2024	22.06.2024	01.07.2024	14.07.2024	Відпов

Рисунок В.12 – Відображення таблиці “Main”.

Таблиці звітності

Main Details Measurements

Пошук

	Дата подання бланку	ПІБ	Шифр	Рівень мобільності (1-2-3-4)	Стать	Вік	Зріст	Вага	Нога	Вражена кінцівка	Майстер	Термін виконання	Ціна
▶	01.01.2024	Ковален...	101	3	Ч	45	180 см	75 кг	ліва	гомілка	Іванов І.І.	28.01.2024	5000
	15.01.2024	Петренк...	102	2	Ж	50	165 см	60 кг	права	стегно	Петров П.П.	10.02.2024	6000
	01.02.2024	Сидорен...	103	4	Ч	30	175 см	70 кг	ліва	гомілка	Сидоров С.С.	10.03.2024	7000
	28.02.2024	Іваненк...	104	1	Ж	40	170 см	65 кг	права	стегно	Коваленко К.К.	05.04.2024	5500
	25.03.2024	Гриценк...	105	5	Ч	35	182 см	80 кг	ліва	гомілка	Гриценко Г.Г.	10.05.2024	4800
	01.05.2024	Мельни...	106	3	Ч	55	178 см	85 кг	права	стегно	Мельниченко ...	16.06.2024	6200
	01.06.2024	Шевчен...	107	4	Ж	60	160 см	70 кг	ліва	гомілка	Шевченко Ш.Ш.	15.07.2024	7000

Рисунок В.13 – Відображення таблиці “Details” (form_description).

Таблиці звітності

MainDetailsMeasurements

Пошук

	Шифр	Тип ампутації	Проксимальний замір	Кістковий замір	Медіальний замір	Дистальний замір	Довжина культи	Довжина стопи (здорова нога)	Довжина від підколіна до підлоги	Довжина від тубера до підлоги
▶	101	гомілка	20	18	16	14	30	25	40	70
	102	стегно	22	20	18	16	32	26	42	78
	103	гомілка	24	22	20	18	34	27	44	72
	104	стегно	26	24	22	20	36	28	46	70
	105	гомілка	28	26	24	22	38	29	48	72
	106	стегно	30	28	26	24	40	30	50	71
	107	гомілка	32	30	28	26	42	31	52	74

Рисунок В.14 – Відображення таблиці “Measurements”.

Таблиці звітності

MainDetailsMeasurements

Пошук Коваленко

	Шифр	Майстер	Етап 1 Розміри	Етап 2 Замовлення комплектуючих	Етап 3 Зняття негативу	Етап 4 Формування позитиву	Етап 5 Ламінування	Етап 6 Збірка	Етап 7 Видача пацієнту	Етап 8 Зауваження
▶	104	Ковален...	Виконано	Виконано	03.03.2024	09.03.2024	18.03.2024	25.03.2024	06.04.2024	Додатково

Рисунок В.15 – Робота пошуку по таблиці “Main”.

Таблиці звітності

MainDetailsMeasurements

Пошук 101

	Дата подання бланку	ПІБ	Шифр	Рівень мобільності (1-2-3-4)	Стать	Вік	Зріст	Вага	Нога	Вражена кінцівка	Майстер	Термін виконання	Ціна
▶	01.01.2024	Ковален...	101	3	Ч	45	180 см	75 кг	ліва	гомілка	Іванов І.І.	28.01.2024	5000

Рисунок В.16 – Робота пошуку по таблиці “Details”.

Таблиці звітності

Main Details Measurements

Пошук

	Шифр	Тип ампутації	Проксимальний замір	Кістковий замір	Медіальний замір	Дистальний замір	Довжина культи	Довжина стопи (здорова нога)	Довжина від підколіна до підлоги	Довжина від тубера до підлоги
▶	101	гомілка	20	18	16	14	30	25	40	70
	103	гомілка	24	22	20	18	34	27	44	72
	105	гомілка	28	26	24	22	38	29	48	72
	107	гомілка	32	30	28	26	42	31	52	74

Рисунок В.17 – Робота пошуку по таблиці “Measurements”

Form1.Designer.cs **Conect_Class.cs*** Get_DataSQL.cs Form1.cs Form1.cs [Design]

medicine medicine.Conect_Class

```

1 using System;
2 using System.Collections.Generic;
3 using System.Data;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7 using System.Windows.Forms;
8 using MySql.Data.MySqlClient;
9
10 namespace medicine
11 {
12     17 references
13     internal class Conect_Class
14     {
15         static string DBConnect = "server=127.0.0.1; user=root; database=medicine; password=";
16         static public MySqlDataAdapter msDataAdapter;
17         static MySqlConnection myconnect;
18         static public MySqlCommand msCommand;
19
20         1 reference
21         public static bool ConnectionDB()
22         {
23             try
24             {
25                 myconnect = new MySqlConnection(DBConnect);
26                 myconnect.Open();
27                 msCommand = new MySqlCommand();
28                 msCommand.Connection = myconnect;
29                 msDataAdapter = new MySqlDataAdapter(msCommand);
30                 return true;
31             }
32             catch
33             {
34                 MessageBox.Show("Помилка підключення з базов даних", "Помилка!", MessageBoxButtons.OK, MessageBoxIcon.Error);
35                 return false;
36             }
37         }
38
39         0 references
40         public static void CloseDB()
41         {
42             if (myconnect != null && myconnect.State == ConnectionState.Open)
43             {
44                 myconnect.Close();
45             }
46         }
47
48         2 references
49         public static MySqlConnection getConnection()
50         {
51             return myconnect;
52         }
53     }
54 }

```

Рисунок В.18 – Код класу “Conect_Class”.

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Data;
6  using System.Threading.Tasks;
7  using System.Windows.Forms;
8  using System.Data.Common;
9  using MySql.Data.MySqlClient;
10
11 namespace medicine
12 {
13     internal class Get_DataSQL
14     {
15         static public DataTable dtLoad = new DataTable();
16         static public DataTable dtLoad1 = new DataTable();
17         static public DataTable dtLoad2 = new DataTable();
18
19         //Main
20         static public void GetMain()
21         {
22             try
23             {
24                 EnsureConnected();
25                 Connect_Class.mCommand.CommandText = "SELECT * FROM main";
26                 dtLoad.Clear();
27                 Connect_Class.mDataAdapter.SelectCommand = Connect_Class.mCommand;
28                 Connect_Class.mDataAdapter.Fill(dtLoad);
29             }
30             catch (Exception ex)
31             {
32                 Console.WriteLine("An error occurred: " + ex.Message);
33             }
34         }
35
36         //FormDesc
37         static public void GetForm_desc()
38         {
39             try
40             {
41                 EnsureConnected();
42                 Connect_Class.mCommand.CommandText = "SELECT * FROM form_description";
43                 dtLoad1.Clear();
44                 Connect_Class.mDataAdapter.SelectCommand = Connect_Class.mCommand;
45                 Connect_Class.mDataAdapter.Fill(dtLoad1);
46             }
47             catch (Exception ex)
48             {
49                 Console.WriteLine("An error occurred: " + ex.Message);
50             }
51         }
52
53         //Measurements
54         static public void GetMeasurements()
55         {
56             try
57             {
58                 EnsureConnected();
59                 Connect_Class.mCommand.CommandText = "SELECT * FROM measurements";
60                 dtLoad2.Clear();
61                 Connect_Class.mDataAdapter.SelectCommand = Connect_Class.mCommand;
62                 Connect_Class.mDataAdapter.Fill(dtLoad2);
63             }
64             catch (Exception ex)
65             {
66                 Console.WriteLine("An error occurred: " + ex.Message);
67             }
68         }
69
70         //EnsureConnected
71         private static void EnsureConnected()
72         {
73             if (Connect_Class.mCommand == null || Connect_Class.mDataAdapter == null || Connect_Class.getConnection() == null || Connect_Class.getConnection().State != ConnectionState.Open)
74             {
75                 if (!Connect_Class.ConnectionOK())
76                 {
77                     throw new InvalidOperationException("Не удалось подключиться по базе данных.");
78                 }
79             }
80         }
81     }
82 }

```

Рисунок В.19 – Код класу “Get_DataSQL”.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using MySql.Data.MySqlClient;
using System.Data.SqlClient;
using System.Windows.Forms;
using System.Data;
using System.Xml.Linq;
using System.Net;
using Org.BouncyCastle.Utilities.Net;

namespace medicine
{
    1 reference
    public partial class Form1 : Form
    {
        1 reference
        public Form1()
        {
            InitializeComponent();
            TextBoxMain.TextChanged += TextBoxMain_TextChanged;
        }

        1 reference
        private void Form1_Load(object sender, EventArgs e)
        {
            Get_DataSQL.GetMain();
            dataGridviewMain.DataSource = Get_DataSQL.dtLoad;
            Get_DataSQL.GetForm_Descr();
            dataGridviewDetails.DataSource = Get_DataSQL.dtLoad1;
            Get_DataSQL.GetMeasurements();
            dataGridviewMeasurements.DataSource = Get_DataSQL.dtLoad2;
        }

        2 reference
        private void TextBoxMain_TextChanged(object sender, EventArgs e)
        {
            string searchText = TextBoxMain.Text.Trim();

            DataView dv = new DataView(Get_DataSQL.dtLoad);

            string filterExpression = "";

            foreach (DataColumn column in dv.Table.Columns)
            {
                if (!string.IsNullOrEmpty(filterExpression))
                {
                    filterExpression += " OR ";
                }
                filterExpression += $"*CONVERT([{column.ColumnName}], 'System.String') LIKE '{searchText}%'";
            }

            dv.RowFilter = filterExpression;

            dataGridviewMain.DataSource = dv.ToTable();
        }
    }
}

```

Рисунок В.20 – Код до “Form1”.

```

Form1.Designer.cs  Conect_Class.cs  Get_DataSQL.cs  Form1.cs
C# medicine
12
13     foreach (DataColumn column in dv.Table.Columns)
14     {
15         if (!string.IsNullOrEmpty(filterExpression))
16         {
17             filterExpression += " OR ";
18         }
19         filterExpression += $"*CONVERT([{column.ColumnName}], 'System.String') LIKE '%{searchText}%';";
20     }
21
22     dv.RowFilter = filterExpression;
23
24     dataGridViewMain.DataSource = dv.ToTable();
25
26 }
27
28 //reference
29 private void TextBoxDetails_TextChanged(object sender, EventArgs e)
30 {
31     string searchText = TextBoxDetails.Text.Trim();
32     DataView dv = new DataView(Get_DataSQL.dlLoad1);
33     string filterExpression = "";
34
35     foreach (DataColumn column in dv.Table.Columns)
36     {
37         if (!string.IsNullOrEmpty(filterExpression))
38         {
39             filterExpression += " OR ";
40         }
41         filterExpression += $"*CONVERT([{column.ColumnName}], 'System.String') LIKE '%{searchText}%';";
42     }
43
44     dv.RowFilter = filterExpression;
45
46     dataGridViewDetails.DataSource = dv.ToTable();
47 }
48
49 //reference
50 private void TextBoxMeasure_TextChanged(object sender, EventArgs e)
51 {
52     string searchText = TextBoxMeasure.Text.Trim();
53     DataView dv = new DataView(Get_DataSQL.dlLoad2);
54     string filterExpression = "";
55
56     foreach (DataColumn column in dv.Table.Columns)
57     {
58         if (!string.IsNullOrEmpty(filterExpression))
59         {
60             filterExpression += " OR ";
61         }
62         filterExpression += $"*CONVERT([{column.ColumnName}], 'System.String') LIKE '%{searchText}%';";
63     }
64
65     dv.RowFilter = filterExpression;
66
67     dataGridViewMeasurements.DataSource = dv.ToTable();
68 }
69
70 //reference
71 private void Label1_Click(object sender, EventArgs e)
72 {
73 }
74
75 }
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111

```

Рисунок В.21 – Код до “Form1”.

ДОДАТОК Г

(довідковий)

Лістинг програми Prosthetic Manager

ClientCare Pro Prosthetic Database Manager

```
//Клас підключення до бази даних phpMyAdmin
internal class Conect_Class
{
    static string DBConnect = "server=127.0.0.1; user=root; database=medicine;
password=";
    static public MySqlDataAdapter msDataAdapter;
    static MySqlConnection myconnect;
    static public MySqlCommand msCommand;

    public static bool ConnectionDB()
    {
        try
        {
            myconnect = new MySqlConnection(DBConnect);
            myconnect.Open();
            msCommand = new MySqlCommand();
            msCommand.Connection = myconnect;
            msDataAdapter = new MySqlDataAdapter(msCommand);
            return true;
        }
        catch
        {
            MessageBox.Show("Помилка підключення з базою даних", "Помилка!",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
            return false;
        }
    }

    public static void CloseDB()
    {
        if (myconnect != null && myconnect.State == ConnectionState.Open)
        {
            myconnect.Close();
        }
    }

    public static MySqlConnection getConnection()
    {
        return myconnect;
    }
}

//Клас для відображення даних з бази даних phpMyAdmin
internal class Get_DataSQL
{
    static public DataTable dtLoad = new DataTable();
    static public DataTable dtLoad1 = new DataTable();
    static public DataTable dtLoad2 = new DataTable();

    static public void GetMain()
    {
        try
        {
            EnsureConnected();
            Conect_Class.msCommand.CommandText = "SELECT * FROM main";
            dtLoad.Clear();
        }
    }
}
```

```

        Conect_Class.msDataAdapter.SelectCommand = Conect_Class.msCommand;
        Conect_Class.msDataAdapter.Fill(dtLoad);
    }
    catch (Exception ex)
    {
        Console.WriteLine("An error occurred: " + ex.Message);
    }
}

static public void GetForm_Descr()
{
    try
    {
        EnsureConnected();
        Conect_Class.msCommand.CommandText = "SELECT * FROM form_description";
        dtLoad1.Clear();
        Conect_Class.msDataAdapter.SelectCommand = Conect_Class.msCommand;
        Conect_Class.msDataAdapter.Fill(dtLoad1);
    }
    catch (Exception ex)
    {
        Console.WriteLine("An error occurred: " + ex.Message);
    }
}

static public void GetMeasurements()
{
    try
    {
        EnsureConnected();
        Conect_Class.msCommand.CommandText = "SELECT * FROM measurements";
        dtLoad2.Clear();
        Conect_Class.msDataAdapter.SelectCommand = Conect_Class.msCommand;
        Conect_Class.msDataAdapter.Fill(dtLoad2);
    }
    catch (Exception ex)
    {
        Console.WriteLine("An error occurred: " + ex.Message);
    }
}

private static void EnsureConnected()
{
    if (Conect_Class.msCommand == null || Conect_Class.msDataAdapter == null ||
        Conect_Class.getConnection() == null || Conect_Class.getConnection().State !=
        ConnectionState.Open)
    {
        if (!Conect_Class.ConnectionDB())
        {
            throw new InvalidOperationException("Не вдалося підключитися до
бази даних.");
        }
    }
}

//Логіка роботи WindowsForms програми:
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
        TextBoxMain.TextChanged += TextBoxMain_TextChanged;
    }
}

```

```

private void Form1_Load(object sender, EventArgs e)
{
    Get_DataSQL.GetMain();
    dataGridViewMain.DataSource = Get_DataSQL.dtLoad;
    Get_DataSQL.GetForm_Descr();
    dataGridViewDetails.DataSource = Get_DataSQL.dtLoad1;
    Get_DataSQL.GetMeasurements();
    dataGridViewMeasurements.DataSource = Get_DataSQL.dtLoad2;
}

private void TextBoxMain_TextChanged(object sender, EventArgs e)
{
    string searchText = TextBoxMain.Text.Trim();

    DataView dv = new DataView(Get_DataSQL.dtLoad);

    string filterExpression = "";

    foreach (DataColumn column in dv.Table.Columns)
    {
        if (!string.IsNullOrEmpty(filterExpression))
        {
            filterExpression += " OR ";
        }
        filterExpression += $"CONVERT([{column.ColumnName}], 'System.String')
LIKE '%{searchText}%'";
    }

    dv.RowFilter = filterExpression;

    dataGridViewMain.DataSource = dv.ToTable();
}

private void TextBoxDetails_TextChanged(object sender, EventArgs e)
{
    string searchText = TextBoxDetails.Text.Trim();

    DataView dv = new DataView(Get_DataSQL.dtLoad1);

    string filterExpression = "";

    foreach (DataColumn column in dv.Table.Columns)
    {
        if (!string.IsNullOrEmpty(filterExpression))
        {
            filterExpression += " OR ";
        }
        filterExpression += $"CONVERT([{column.ColumnName}], 'System.String')
LIKE '%{searchText}%'";
    }

    dv.RowFilter = filterExpression;

    dataGridViewDetails.DataSource = dv.ToTable();
}

private void TextBoxMeasure_TextChanged(object sender, EventArgs e)
{
    string searchText = TextBoxMeasure.Text.Trim();

    DataView dv = new DataView(Get_DataSQL.dtLoad2);

    string filterExpression = "";

    foreach (DataColumn column in dv.Table.Columns)

```

```

        {
            if (!string.IsNullOrEmpty(filterExpression))
            {
                filterExpression += " OR ";
            }
            filterExpression += $"CONVERT([{column.ColumnName}], 'System.String')
LIKE '%{searchText}%'";
        }

        dv.RowFilter = filterExpression;

        dataGridViewMeasurements.DataSource = dv.ToTable();
    }

    private void label2_Click(object sender, EventArgs e)
    {
    }
}

```