

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та
роботехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

Розроблення програмного забезпечення для автоматизованого обліку
конструкторсько-технологічної документації
комп'ютерно-інтегрованого виробництва
(тема)

Виконав:
студент 2 курсу, групи КІТПВм-22-1
Шмига Р. В.
(прізвище, ініціали)

Спеціальності 151 Автоматизація та
комп'ютерно-інтегровані технології
(код і повна назва спеціальності)

Тип програми Освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютерно-
інтегровані технологічні процеси і
виробництва
(повна назва освітньої програми)

Керівник доц. Аллахверанов Р. Ю.
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри КІТАР

_____ (підпис)

Невлюдов І. Ш.
(прізвище, ініціали)

2024р.

Харківський національний університет радіоелектроніки

Факультет	Автоматики і комп'ютеризованих технологій
Кафедра	Комп'ютерно-інтегрованих технологій, автоматизації та роботехніки
Рівень вищої освіти	другий (магістерський)
Спеціальність	151 Автоматизація та комп'ютерно-інтегровані технології
Тип програми	освітньо-професійна
Освітня програма	Комп'ютерно-інтегровані технологічні процеси і виробництва

(код і повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри КІТАР

(підпис)

« ____ » _____ 20 ____ р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Студентові

Шмизі Родіону Володимировичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Розроблення програмного забезпечення для автоматизованого обліку конструкторсько-технологічної документації комп'ютерно-інтегрованого виробництва*

затверджена наказом по університету від *“03” листопада 2023р. №1287 Ст.*

2. Термін подання студентом роботи *“ 24 ” січня 2024р.*

3. Вихідні дані до роботи *3.1 Основні функції – облік ТМЦ, конструкторської та технологічної документації на комп'ютерно-інтегрованому виробництві;*

3.2 Середовище розробки програмного засобу – MS Visual Studio, мова C#;

3.3 Інтерфейс програми – графічний;

3.4 Тип бази даних – PostgreSQL;

3.5 Тип операційної системи – Microsoft Windows;

3.6 Оформлення текстової документації згідно ДСТУ 3008–2015.

4. Перелік питань, що потрібно опрацювати в роботі *4.1 Вступ;*

4.2 Аналіз літератури за темою досліджень;

4.3 Розробка архітектури та моделювання роботи програми;

4.4 Розробка структури бази даних;

4.5 Розробка програми для автоматизації обліку конструкторсько-технологічної документації;

4.6 Забезпечення безпечних умов праці при проведенні досліджень;

4.7 Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій Демонстраційний матеріал представлений у форматі презентації PowerPoint (*.ppt) – 14 с. формату А4

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	<i>Аналіз літератури за темою досліджень</i>	<i>03.11 – 09.11.23</i>	<i>виконано</i>
2	<i>Розробка архітектури та моделювання роботи програми</i>	<i>10.11 – 20.11.23</i>	<i>виконано</i>
3	<i>Розробка структури бази даних</i>	<i>21.11 – 10.12.23</i>	<i>виконано</i>
4	<i>Розробка програми для автоматизації обліку конструкторсько–технологічної документації</i>	<i>11.12 – 30.12.23</i>	<i>виконано</i>
5	<i>Забезпечення безпечних умов праці при проведенні досліджень</i>	<i>01.01 – 09.01.24</i>	<i>виконано</i>
6	<i>Оформлення пояснювальної записки</i>	<i>10.01 – 13.01.24</i>	<i>виконано</i>
7	<i>Подання роботи на перевірку Інтернет–сервісом Unichesk</i>	<i>14.01 – 16.01.24</i>	<i>виконано</i>
8	<i>Подання роботи на рецензію</i>	<i>17.01 – 19.01.24</i>	<i>виконано</i>
9	<i>Подання роботи на підпис зав. кафедри</i>	<i>20.01 – 22.01.24</i>	<i>виконано</i>
10	<i>Подання кваліфікаційної роботи в ЕК</i>	<i>23.01.24</i>	<i>виконано</i>

Дата видачі завдання 03 жовтня 2023 р.

Студент _____ Шмига Р. В.
(підпис)

Керівник роботи _____ доц. Аллахверанов Р. Ю.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 82 с., 28 рис., 13 табл., 2 додатка, 16 джерела.

ПРОГРАМНИЙ МОДУЛЬ, БАЗА ДАНИХ, АРХІТЕКТУРА ПРОГРАМНОГО МОДУЛЯ, КОМП'ЮТЕРНО-ІНТЕГРОВАНЕ ВИРОБНИЦТВО, ТЕХНОЛОГІЧНА ПІДГОТОВКА ВИРОБНИЦТВА, КОНСТРУКТОРСЬКО-ТЕХНОЛОГІЧНА ДОКУМЕНТАЦІЯ, ТОВАРНО-МАТЕРІАЛЬНІ ЦІННОСТІ.

Об'єкт розробки – автоматизація технологічної підготовки комп'ютерно-інтегрованого виробництва.

Предмет розробки – база даних конструкторсько-технологічної документації комп'ютерно-інтегрованого виробництва.

Мета роботи – розроблення програмного модуля для контролю виконання замовлень створення, редагування та обліку конструкторсько-технологічної документації на комп'ютерно-інтегрованому виробництві.

Проаналізовано технічне завдання, визначені основні вимоги програми, що розробляється. Проведено аналіз загальних принципів організації технологічного процесу виготовлення виробу. Визначені основні вимоги до єдиного інформаційного простору технологічної підготовки виробництва. Проведено аналіз аналогічних програмних модулів. Розроблено архітектуру програмного модуля та виконано моделювання його роботи за допомогою мови UML. Виконано моделювання структури даних в автоматизованому середовищі PowerDesigner. Проведено заходи і розрахунки для забезпечення умов безпечної праці у лабораторії, де виконувались дослідження з кваліфікаційної роботи.

ABSTRACT

Explanatory note: 82 p., 28 figures, 13 tables, 2 supplement, 16 sources.

SOFTWARE MODULE, DATABASE, SOFTWARE MODULE ARCHITECTURE, COMPUTER-INTEGRATED MANUFACTURING, TECHNOLOGICAL PRODUCTION PREPARATION, DESIGN AND TECHNOLOGICAL DOCUMENTATION, GOODS AND MATERIAL VALUES.

The object of development is the automation of technological preparation of computer-integrated production.

The subject of development is a database of design and technological documentation of computer-integrated production.

The purpose of the work is to develop a software module to control the execution of orders for the creation, editing and accounting of design and technological documentation in computer-integrated production.

The technical task was analyzed, the main requirements of the program being developed were determined. The analysis of the general principles of the organization of the technological process of manufacturing the product was carried out. The main requirements for a unified information space for technological preparation of production are defined. Similar software modules were analyzed. The architecture of the software module was developed and its work was modeled using the UML language. Modeling of the data structure was performed in the PowerDesigner automated environment. Measures and calculations were carried out to ensure safe working conditions in the laboratory, where studies on qualification work were carried out.

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

«22» січня 2024 р.



Шмига Р. В.

ЗМІСТ

Перелік скорочень	8
Вступ	9
1 Аналіз технологічного процесу сучасних виробництв	11
1.1 Загальні принципи організації технологічного процесу виготовлення продукції	11
1.2 Аналіз етапів виготовлення продукції	13
1.3 Вимоги до автоматизованих систем технологічної підготовки виробництва	17
1.4 Аналіз аналогічних програмних модулів	23
2 Моделювання роботи програмного модуля	30
2.1 Розробка архітектури програмного модуля	30
2.2 Моделювання поведінки програмного модуля	32
3 Розробка структури бази даних	39
3.1 Обирання системи керування базами даних	39
3.2 Структури сутності бази даних	42
4 Розробка програми для автоматизації обліку конструкторсько–технологічної документації	54
4.1 Вибір мови середовища програмування	54
4.2 Розробка графічної оболонки програми	55
4.3 Розробка запитів до бази даних	67
4.4 Оцінка складності програмного засобу	69
4.5 Забезпечення безпечних умов праці у лабораторій при проведенні досліджень	76
Висновки	79
Перелік джерел посилання	81
Додаток А Апробація результатів досліджень	83
Додаток Б Демонстраційний матеріал	93

ПЕРЕЛІК СКОРОЧЕНЬ

АТПВКІВ – автоматизація технологічної підготовки комп'ютерно–інтегрованого виробництва;

БД – база даних;

ЕОМ – електронно–обчислювальна машина;

ІМС – інтегральні мікросхеми;

ІС – інтегральні схеми;

КПО – коефіцієнт природної освітленості;

КТД – конструкторсько–технологічна документація;

ЄІП – єдиний інформаційний простір;

ПЗ – програмне забезпечення;

СКБД – система керування базами даних;

ТМЦ – товарно–матеріальні цінності;

ТП – технологічний процес;

ТПВ – технологічна підготовка виробництва;

ERP – Enterprise Resource Planning;

PDM – Product Data management.

ВСТУП

Технологічна підготовка виробництва (ТПВ) на комп'ютерно–інтегрованому виробництві є одним з основних етапів створення промислових виробів. Рівень ТПВ сьогодні визначає терміни випуску нової продукції, її якість і конкурентоспроможність. Тому особливої актуальності набувають питання автоматизації ТПВ.

Необхідність підвищення ефективності ТПВ пояснюється тим, що проектування технологічної документації в більшості випадків значно (від двох до п'яти разів) перевершує трудомісткість розробки конструкторської документації. Відчутне підвищення ефективності ТПВ в порівнянні з її існуючим рівнем можливо тільки при виконанні наступних умов:

- наявність єдиного інформаційного простору для фахівців конструкторської та технологічної служб підприємства;
- підвищення швидкості розробки і обґрунтованості планів ТПВ, безперервний контроль їх виконання;
- змістовна допомога співробітникам технологічної служби підприємства протягом всіх етапів розробки технологічної документації на основі застосування математичних методів та ідей штучного інтелекту, використання сучасних засобів обчислювальної техніки.

Перераховані умови реалізуються в автоматизації технологічної підготовки комп'ютерно–інтегрованого виробництва (АТПКІВ). АТПКІВ являє собою апаратно–програмну систему, однак її апаратна і програмна частини не є рівноцінними. Ядром АТПКІВ будь–якого підприємства є програмне забезпечення цієї системи. В першу чергу саме воно забезпечує виконання даних умов. Зазначена обставина пояснюється тим, що в програмному забезпеченні акумулюються знання найбільш досвідчених фахівців і вчених у галузі технології машинобудування. В першу чергу використання таких знань і забезпечує необхідну підвищення ефективності ТПВ.

Об'єкт розробки – автоматизація технологічної підготовки комп'ютерно–інтегрованого виробництва.

Предмет розробки – база даних конструкторсько–технологічної документації комп'ютерно–інтегрованого виробництва.

Мета роботи – розроблення програмного модуля для контролю виконання замовлень створення, редагування та обліку конструкторсько–технологічної документації на комп'ютерно–інтегрованому виробництві.

Для досягнення даної мети необхідно вирішити такі завдання:

- проаналізувати технічне завдання, визначити основні вимоги програми, що розробляється;

- провести аналіз загальних принципів організації технологічного процесу виготовлення виробу;

- визначити основні вимоги до єдиного інформаційного простору технологічної підготовки виробництва;

- провести аналіз аналогічних програмних модулів;

- розробити архітектуру програмного модуля та виконати моделювання його роботи за допомогою мови UML.

- виконати моделювання структури даних в автоматизованому середовищі PowerDesigner;

- провести заходи і розрахунки для забезпечення умов безпечної праці у лабораторії, де виконувались дослідження з кваліфікаційної роботи.

Робота виконується згідно з [1–4], як складова наукових досліджень, які здійснюються на кафедрі КІТАР Харківського національного університету радіоелектроніки, результати дослідження опубліковані у [5].

1 АНАЛІЗ ТЕХНОЛОГІЧНОГО ПРОЦЕСУ СУЧАСНИХ ВИРОБНИЦТВ

1.1 Аналіз загальних принципів організації технологічного процесу виготовлення виробу

Технологічний процес (ТП) є складним взаємозв'язком різних механізмів, ресурсів, характеристик і показників. Для найбільш ефективної роботи виробничого підприємства потрібно налагодити технологічний процес так, щоб в результаті отримати раціональне поєднання праці робітників із речовими елементами виробництва.

Поділ праці обов'язково передбачає об'єднання, тому що кожна часткова робота набуває певного сенсу лише в поєднанні з іншими частковими роботами (рисунок 1.1). Отже, спеціалізація праці отримує своє доповнення в її кооперуванні. Таким чином, принципи організації технологічного процесу повинні ґрунтуватись на внутрішній декомпозиції взаємопов'язаних виробництв.

Технологічне проектування полягає, перш за все, у виборі найбільш економічного для даних конкретних умов методу отримання заготовок і деталей, а також встановленні раціональної послідовності операцій обробки, призначення необхідних знарядь виробництва та регламентації їхнього застосування, визначенні трудомісткості та собівартості виготовлення виробів. Технологічний процес необхідно розробити таким чином, щоб обладнання, інструменти, пристосування, сировина, виробничі площі використовувалися максимально повно, раціонально та правильно. Такий підхід дозволить виконати виробничий план з найменшими матеріальними витратами.

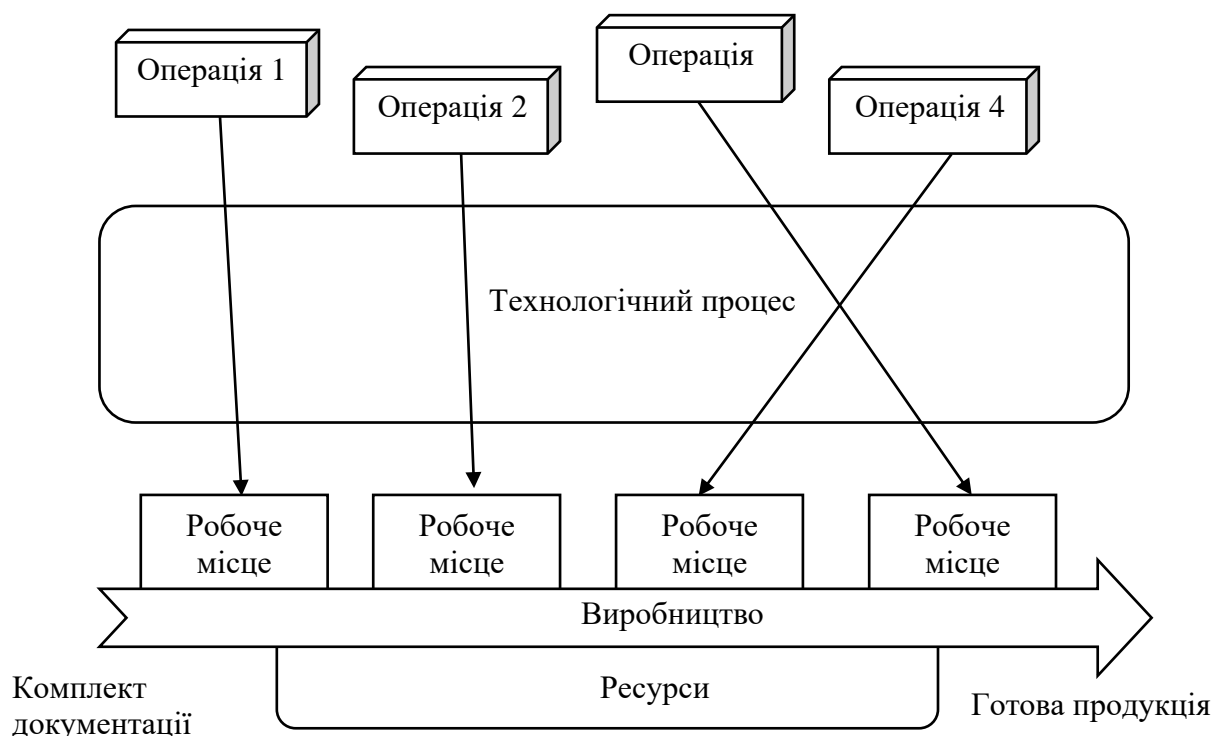


Рисунок 1.1 – Виробництво продукції на підприємстві

На виробничих підприємствах за основу виробництва тих чи інших виробів узято потокову лінію. Іншими словами, це сукупність машинних (верстати, прилади, пристрої) та людських (робоча сила) ресурсів. На вхід надходять сировина з матеріалами, а потім за допомогою виконання певних операцій на наявному обладнанні на виході отримують готову продукцію, тобто вироби та матеріали, процес роботи над якими в даному виробничому циклі повністю завершено (рисунок 1.1).

Оскільки потокова лінія має послідовну структуру розміщення її елементів, відповідно, вихід з ладу одного з них тягне за собою відмову всієї системи. Це призводить до простою виробництва, а також непередбачених економічних втрат, зміщення термінів виконання замовлення, що негативно вплине на репутацію підприємства і може призвести до зниження конкурентоспроможності виробу.

Проте цю проблему можна вирішити шляхом встановлення кількох поточкових ліній, робота яких може спрямовуватись як на виконання одного замовлення, так і на виробництво різних виробів. Зауважимо, що звернути

увагу на властивість взаємозамінності технологій. У першому випадку економічна вигода полягає у зменшенні термінів роботи з виробництва одного виробу, а в іншому – можливості одночасної роботи над двома різними замовленнями. Реалізація такого підходу забезпечується впровадженням систем керування інженерними даними та життєвим циклом виробів (PDM–PLM систем) за допомогою організації єдиного інформаційного простору конструкторсько–технологічної підготовки виробництва [6].

У технологічному процесі всі складові його операції розміщено в часовому просторі, тобто на виконання кожної окремої роботи виділено лімітовану кількість часу. Відповідно тимчасові інтервали між виконанням операцій ($T_1, T_2, T_3 \dots T_n$) також відрізняються між собою (рисунок 1.2).

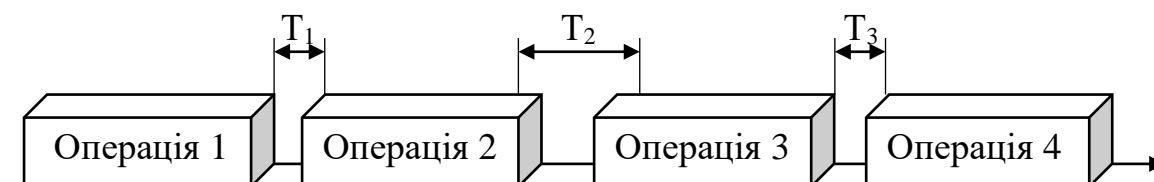


Рисунок 1.2 – Розподілення у часі операцій технологічного процесу

Під час переналагодження виробництва під замовлення клієнта зазвичай необхідно переналаштовувати потокову лінію, змінюючи робочі місця, ресурси, обладнання, матеріали. До того ж, відбуваються зміни за часом виконання вже інших робочих операцій.

1.2 Аналіз етапів виготовлення виробу

Повний життєвий цикл будь–якого виробу за час виготовлення супроводжується великою кількістю різноманітної інформації. Для підвищення ефективності керування виробництвом необхідно інтегрувати автоматизовані процеси та належні до них дані. Питання інтеграції інформаційних ресурсів, без вирішення яких неможливо ефективно підготувати

виробництва, є доволі важливими під час створення єдиного інформаційного простору підприємства. Зауважимо, що найчастіше ця інтеграція базується на паралельному застосуванні PDM (Product Data management) та ERP (Enterprise Resource Planning) систем, одночасно із зовнішніми корпоративними довідниками та класифікаторами [6].

На сьогодні розроблено значну кількість різноманітних систем подібного роду, причому потужні засоби інтеграції цих систем, в тому числі з системами автоматизованого проєктування дозволяють забезпечити вимоги підприємств [4]. Такі системи дозволяють:

- керувати зберіганням даних та документами (організація електронних сховищ даних, керування рівнями версій та змінами);
- керувати потоками робіт, процесами, структурою продукту (визначення та модифікація структури, підтримка версій та опцій дизайну);
- автоматизувати генерацію вибірок і звітів, можливість відображати повідомлення з будь-якою інформацією, що розташована в базі даних;
- забезпечити безпечний доступ до інформації, визначивши права окремих користувачів або їхніх груп.

Єдиний інформаційний простір (ЄІП) ТПВ реалізується через PDM-системи та як платформу використовує мережу автоматизованих робочих місць. Це дозволяє:

- приймати та зберігати проєкт виробу в електронному вигляді;
- ефективно відстежувати поточний стан ТПВ-виробу;
- забезпечити цілісність, несуперечність, а також відсутність дублювання даних;
- організувати швидкий авторизований перегляд усіх моделей та документів;
- забезпечити оперативний обмін інформацією між користувачами АТПВКІВ;
- забезпечити швидке проходження конструкторських та технологічних змін;

- автоматизувати процеси керування потоками виробничих завдань у сфері ТПВ;
- забезпечити інформаційну узгодженість роботи всіх підсистем АТПВКІВ;
- підтримати відкритість АТПВКІВ, зручність в адаптації до мінливих умов виробництва;
- забезпечити інформаційний обмін із системами, що реалізують підтримку різних етапів життєвого циклу виробу.

ЄП ТПВ містить:

- інформацію про обладнання та засоби технологічного оснащення;
- інформацію про технологічні процеси виготовлення виробу;
- інформацію про деталі та складальні одиниці виробу;
- планово–облікову інформацію;
- нормативно–довідкову інформацію;

Життєвий цикл виробу подано на рисунку 1.3.

ЄП – це основа АТПВКІВ, оскільки в його середовищі реалізуються і цільові, і власні функції АТПВКІВ (цільові функції відповідають за завдання, для вирішення яких створюється АТПВКІВ, а власні функції – за завдання, що мають вирішуватися в АТПВКІВ для забезпечення цільових функцій).

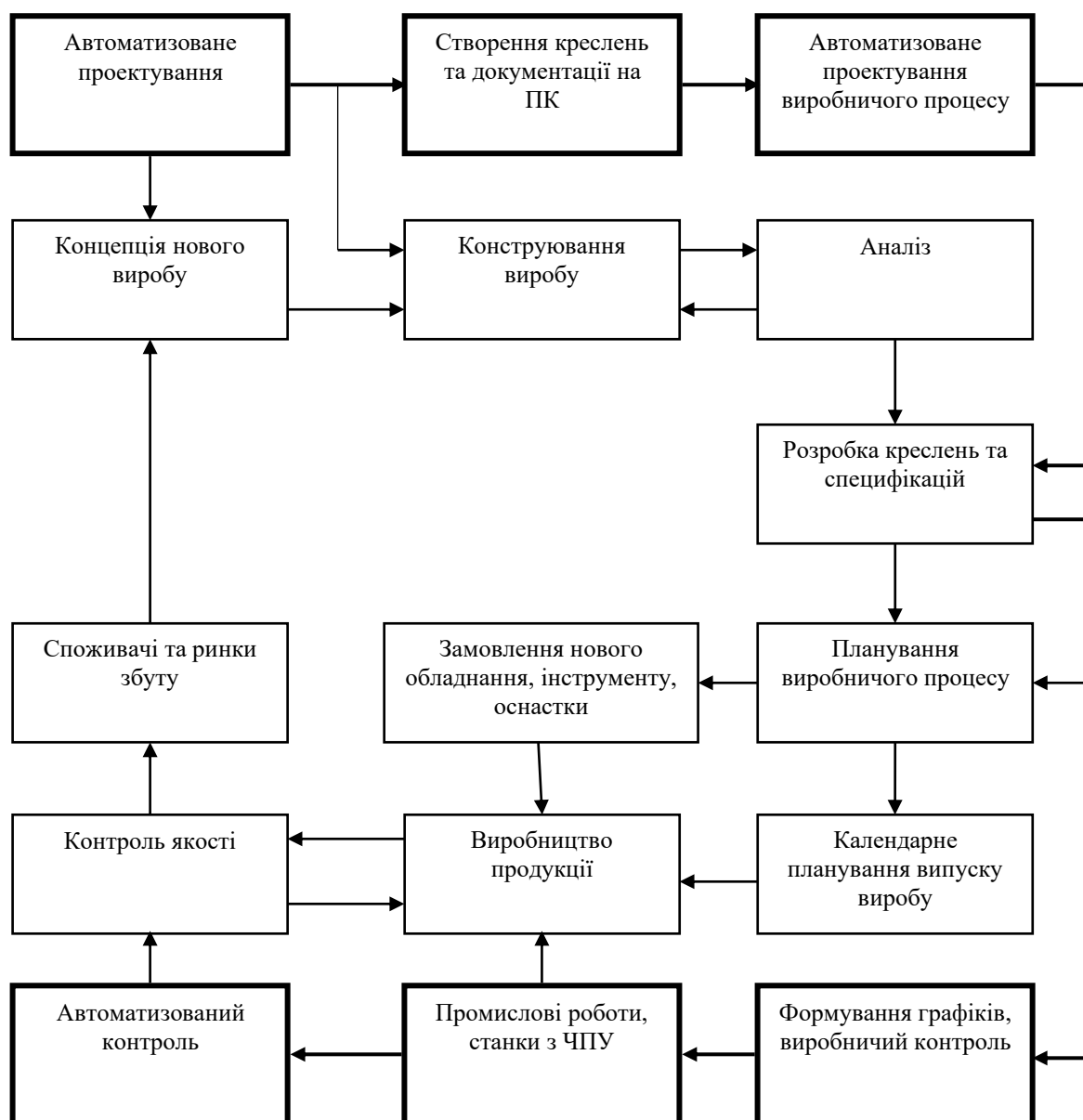


Рисунок 1.3 – Життєвий цикл виробу

АТПВКІВ містить:

- програмну складову: поставку CAD/CAM, CAE і PDM-систем, розробку застосунків, налаштування баз даних тощо. Належить сюди і постачання засобів візуального моделювання бізнес-процесів ТПВ;

- технічну складову: постачання засобів, які утворюють технічну платформу АТПВКІВ (сервери, комп'ютери АРМ, засоби обчислювальних мереж). Така підтримка може реалізовуватися інжиніринговою фірмою за рахунок встановлення партнерських домовленостей із фірмами-постачальниками комп'ютерної техніки;

- інформаційну складову: забезпечує підприємство відомостями, що необхідні для прийняття організаційно–технічних рішень проєкту (порівняльні характеристики різних систем, інформація про хід проєктів АТПВКІВ на інших підприємствах тощо);

- методичну складову: передачу підприємству різних методичних матеріалів щодо реалізації проєкту (зокрема, методику застосування мови UML для моделювання бізнес–процесів, методики розв'язання різних завдань ТІП за допомогою CAD/CAM і CAE–систем, методику побудови єдиної бази даних ТПВ тощо), а також проведення відповідного навчання;

- інженерну складову: допомагає підприємству більш оперативно впроваджувати компоненти АТПВКІВ за рахунок безпосереднього розв'язання інжиніринговою фірмою окремих завдань ТПВ за допомогою постачання програмних систем (проєктування конкретної прес–форми в CAD/CAM–системі, моделювання процесу гарячого штампування заданої деталі в CAE–системі тощо) [6];

- кадрову складову: допомога підприємству з вирішенням питання щодо нестачі фахівців, які необхідні для участі в проєкті АТПВКІВ. Така допомога може виконуватися інжиніринговою фірмою, наприклад, за рахунок тісного контакту (партнерських відносин) з кафедрами зво, що забезпечують підготовку студентів за необхідними спеціальностями.

1.3 Аналіз вимог до автоматизованих систем ТПВ

Головна мета PDM полягає у підтримці електронного опису продукту (виробу) на всіх стадіях його життєвого циклу. Така підтримка повинна забезпечувати розв'язання таких завдань:

- ведення проєктів, зокрема, керування роботами, процедурами та документами проєкту, контроль за виконанням проєкту;

- планування та диспетчеризація робіт;

- розподіл прав доступу до інформації між окремими учасниками проєкту або їхніми групами;

- організація та ведення розподілених архівів конструкторської, технологічної та управлінської документації (електронні архіви);
- керування змінами в документації: контроль за версіями документів, ведення протоколу роботи з документами, листів реєстрації змін та сповіщень;
- фіксування стандартних етапів проходження документів, контроль за цими проходженнями;
- інтеграція з CAD/CAM-системами та їхніми додатками, використаними під час проектування;
- контроль цілісності проєкту;
- пошук необхідної інформації в проєкті відповідно до запитів.

З огляду на її використання великою кількістю фахівців, PDM можна назвати багатокористувацькою системою, що працює в комп'ютерній мережі. Вона організовує єдиний інформаційний простір підприємства, тобто забезпечує створення, зберігання й обробку інформації в єдиній базі даних за допомогою системи керування базами даних (СКБД).

Діяльність АТПВКІВ як єдиної, цілісної системи передбачає функціонування всіх її компонент в єдиному інформаційному просторі (ЄІП) ТПВ. ЄІП називається єдине комп'ютерне інформаційне середовище, що реалізується засобами PDM-системи та забезпечує спільну, узгоджену роботу конструкторів, технологів та інших фахівців підприємства протягом усього часу життєвого циклу виробу.

Значна кількість різних видів даних, які застосовуються в АТПВКІВ, вимагає визначення деяких загальних критеріїв, які повинні задовільняти всі ці дані. Перефразовуючи, необхідно визначити базову модель даних. Тільки в такому разі можна буде побудувати уніфіковані й ефективні механізми зберігання й обробки інформації. Вибираючи базові моделі даних, необхідно враховувати наступне.

АТПВКІВ створюється та використовується значною кількістю фахівців, які розв'язують різні завдання. Отже, модель даних необхідно організувати так, щоб усі фахівці могли ефективно використати результати роботи один одного і, за необхідності, вносити в модель свої зміни.

Процес створення АТПВКІВ є ітеративним, у зв'язку з цим модель, з одного боку, повинна допускати послідовні уточнення, а з іншого – бути стійкою до змін предметної області. Таким чином, за умови модифікації предметної області повинен змінюватися лише певний мінімальний необхідний набір елементів моделі.

До таких вимог доречно використати об'єктно–орієнтований підхід. До об'єктно–орієнтованої моделі даних належать такі основні поняття: об'єкт; характеристики об'єкта; зв'язок між об'єктами; характеристики зв'язку. Характеристиками об'єкта можуть виступати будь–які атрибути, що необхідні для подання певного об'єкта при вирішенні поставлених завдань обробки інформації. Зокрема, інформація основного напису документа може бути характеристикою саме цього документа.

Між об'єктами організуються зв'язки та визначаються їхні характеристики. Звертаємо увагу, що характеристики зв'язків не залежать від характеристик об'єктів, яких вони пов'язують. Розглянемо "Номер зони", "Номер позиції" та "Кількість", які є характеристиками ієрархічної зв'язку деталі зі складальною одиницею. Дана деталь може бути пов'язаною з різними складальними одиницями, проте, зазначені характеристики можуть бути різними. Без встановлення зв'язку дані характеристики не мають сенсу [6].

Отже, кожен виріб або документ є самостійними об'єктами, що можуть бути пов'язані між собою різними видами зв'язків, а вся інформація про об'єкт повинна бути "розкладена" по чотирьох "поличках" об'єктно–орієнтованої моделі: об'єкт, характеристики об'єкта, зв'язок об'єкта з іншими об'єктами; характеристики зв'язку.

Першим і водночас одним із найважливіших і відповідальних кроків під час побудови ЄІП є розробка системи класифікації інформаційних об'єктів. Основна вимога до системи класифікації полягає у відповідності цільовим функціям ТПВ підприємства. Визначення класів об'єктів та їхніх характеристик формується послідовно по мірі створення АТПВКІВ.

Класом називають набір об'єктів, який мають спільні атрибути, причому серед цих атрибутів є хоча б один, який відрізняє об'єкти даного класу від

об'єктів інших класів в моделюванні предметної області. У разі класифікації виробів об'єкти–деталі (утворюють клас "Деталь") мають характеристику "Матеріал". Вона відрізняє ці об'єкти від усіх інших видів виробів за ЕСКД – складальних одиниць, комплексів, стандартних виробів тощо.

Класифікація інформаційних об'єктів може здійснюватися різними способами. Під час побудови системи класів необхідно прагнути до того, щоб:

- за можливістю уникати дублювання інформації;
- спростити роботу користувача з даними;
- врахувати розподіл прав доступу;
- прискорити пошук даних, виконуючи запити.

Розміщення конкретних даних в ЄПІ, а також їхня обробка реалізується в процесі безпосереднього функціонування підсистем АТПВКІВ. Важливою частиною вмісту ЄПІ є опис структури, складу, геометрії та матеріалів виробів як таких компонентів: складу виробу – це дерево проекту; структура – це складальні файли САД–систем; геометрія – це тривимірні моделі та креслення; матеріали – це БД позначень і властивостей матеріалів.

Ще однією важливою частиною вмісту ЄПІ є опис структури та складу технологічних процесів виготовлення виробів. Опис ТП подається як дерево виконуваних технологічних операцій та переходів, крім того, відповідна технологічна документація, що генерується автоматично в потрібних форматах документів.

Інформація, що надходить на етапах ТПВ, зберігається разом з інформацією з інших етапів ЖЦВ в єдиній базі даних, тому використовується як PDM–системою, так і іншими. Всі ці системи та їхні компоненти, що працюють з інформацією, можна класифікувати за так званим "контуром" програмного забезпечення, що містить:

- внутрішньосистемний контур;
- технологічний контур;
- проєктний контур;
- зовнішній контур (рисунок 1.4).



Рисунок 1.4 – Програмні контури та єдина база даних про виріб

Внутрішньосистемний контур – це програмне забезпечення для виконання певних базових функцій:

- редагування структури бази даних та екранних форм;
- введення до бази даних інформації про об'єкти, ієрархічні та логічні зв'язки між об'єктами;
- ведення складу проектів:
- класифікація об'єктів та успадкування інформації за ієрархією класів;
- ведення життєвого циклу документів;
- автоматичне ведення версій документів;
- пошук документів з облікової інформації та логічними зв'язками;
- автоматичне нарощування позначень документів та об'єктів;
- регламентація прав доступу до інформації;
- експорт / імпорт інформації;
- складання графіків виробничих завдань, відстеження їх виконання.

Технологічним контуром називається набір прикладних програм (підсистем, модулів), розроблених за допомогою мови програмування високого рівня та засобів програмного інтерфейсу API. Створені прикладні програми вирішують найрізноманітніші завдання щодо проєктування, керування та документування в АТПВКІВ. До таких завдань належать:

- проєктування технологічних процесів;
- розрахунок потреби в матеріалах і стандартних виробках;
- формування циклограм збірки;
- розв’язування виробів;
- отримання зведених конструкторсько–технологічних документів.

Проектний контур – це безліч CAD/CAM і CAE–систем, і відповідні засоби інтерфейсу (інтеграції) з внутрішньосистемним контуром PDM, які для вирішення своїх завдань використовують фахівці–проєктувальники. За найпростіший приклад інтеграції візьмемо обмін ідентифікаційними даними документа. CAD–системи наділені функцією введення ідентифікаційних даних (наприклад, в основний напис креслення), а PDM–система наділена функцією введення ідентифікаційних даних до паспортів креслень, які зберігаються в базі даних.

Ще один приклад інтеграції полягає в обміні 3D–моделями. Моделі, що створені в CAD–системі, повинні передаватися до PDM–системи для зберігання, а моделі, що збережені в PDM–системі (на кшталт моделей стандартних виробів або запозичених деталей), повинні передаватися до CAD–системи під час створення моделей складальних одиниць.

Зовнішнім контуром називається споживання інформації, що було створено на етапі ТПВ, які обслуговують інші етапи ЖЦВ (постачання, маркетинг, виробництво, контроль, монтаж, упаковка, реалізація, техобслуговування, утилізація).

Отже, єдиний інформаційний простір створює основу для успішного комплексного функціонування АТПВКІВ, для ефективно організації спільної, узгодженої роботи конструкторів, технологів та інших фахівців ТПВ підприємства. Завдяки цьому здійснюється:

- прискорення процесів ТПВ шляхом паралельного виконання робіт та електронного обміну даними між фахівцями;
- підвищення якості та достовірності інформації шляхом прозорості системи та взаємоконтролю учасників процесів проектування;
- накопичення, а також збереження інформації в електронному вигляді;
- відсутність непотрібного дублювання інформації;
- гнучкість створюваної АТПВКІВ, крім того, зручність її розвитку та адаптації до мінливих умов виробництва.

1.4 Аналіз аналогічних програмних модулів

Серед популярних у світі PDM–систем, які відповідають сучасним вимогам, одне з провідних місць посідає PDM SmarTeam. Система складається з таких основних компонентів:

- SmarTeam є базовою системою, що надає повний набір засобів для спільної роботи щодо створення, редагування, пошуку та зберігання будь–яких типів даних і документів. Забезпечується керування проектами, ведення версій, експорт та імпорт інформації. Для редагування структур баз даних, а також налаштування системи використовується додаток Data Model Designer; для розширення функцій користувача – спеціальний програмний інтерфейс API;
- SmartView є модулем, який забезпечує перегляд векторних, растрових, офісних файлів більше 200 форматів, а також внесення позначок до документів (Redlining);
- SmartVault є компонентом серверної архітектури SmarTeam, який забезпечує захист даних, розподіл прав і контроль доступу до всіх даних і документів;
- SmartFlow є підсистемою маршрутизації даних і документів; забезпечує їхнє автоматичне проходження по підприємству відповідно до заданих процедур узгодження, затвердження, внесення змін тощо;

- SmartWeb є підсистемою, що забезпечує віддалений доступ фахівців до бази даних SmarTeam;

- SmartGateway є підсистемою, що забезпечує інтеграцію з ERP-системами (системами керування виробництвом) та іншими PDM-системами;

- Smart Multi-site є підсистемою, що організовує роботу мережі філій підприємства в єдиному інформаційному просторі;

- MySmarTeam, mySmartPublish, SmartBOM, SmartBriefcase є підсистемами, що реалізують інформаційну інтеграцію підприємства з замовниками та постачальниками.

- SniarTeam забезпечує прийом інформації, що створюється на різних етапах ЖЦВ, причому введення інформації може виконуватися як у системах проєктування, так і в самій PDM. Зберігання інформації виконується через бази даних відомих СКБД (Oracle, InterBase, MS SQL-Server).

Засоби, що дозволяють утворювати структури баз даних, а також екранні форми подання інформації в інтерактивному режимі, без застосування мов програмування, дозволяють легко адаптувати SniarTeam до умов підприємства. Користувачі можуть формувати бази даних стандартних і типових деталей, матеріалів, які використовуються, складів устаткування тощо. Розробка програм для розв'язання різних завдань АТПВКІВ в середовищі PDM SmarTeam здійснюється за допомогою спеціального програмного інтерфейсу API (Application Programming Interface).

Реалізація загальної бази даних про виріб дозволяє організувати процес паралельного проєктування, коли кожен фахівець використовує дані про виріб для розв'язання своїх завдань. Навіть у тих випадках, коли наступний проєктант застосовує результати роботи попереднього, використання паралельного проєктування може помітно знизити загальний час ТПВ (рисунок 1.5).

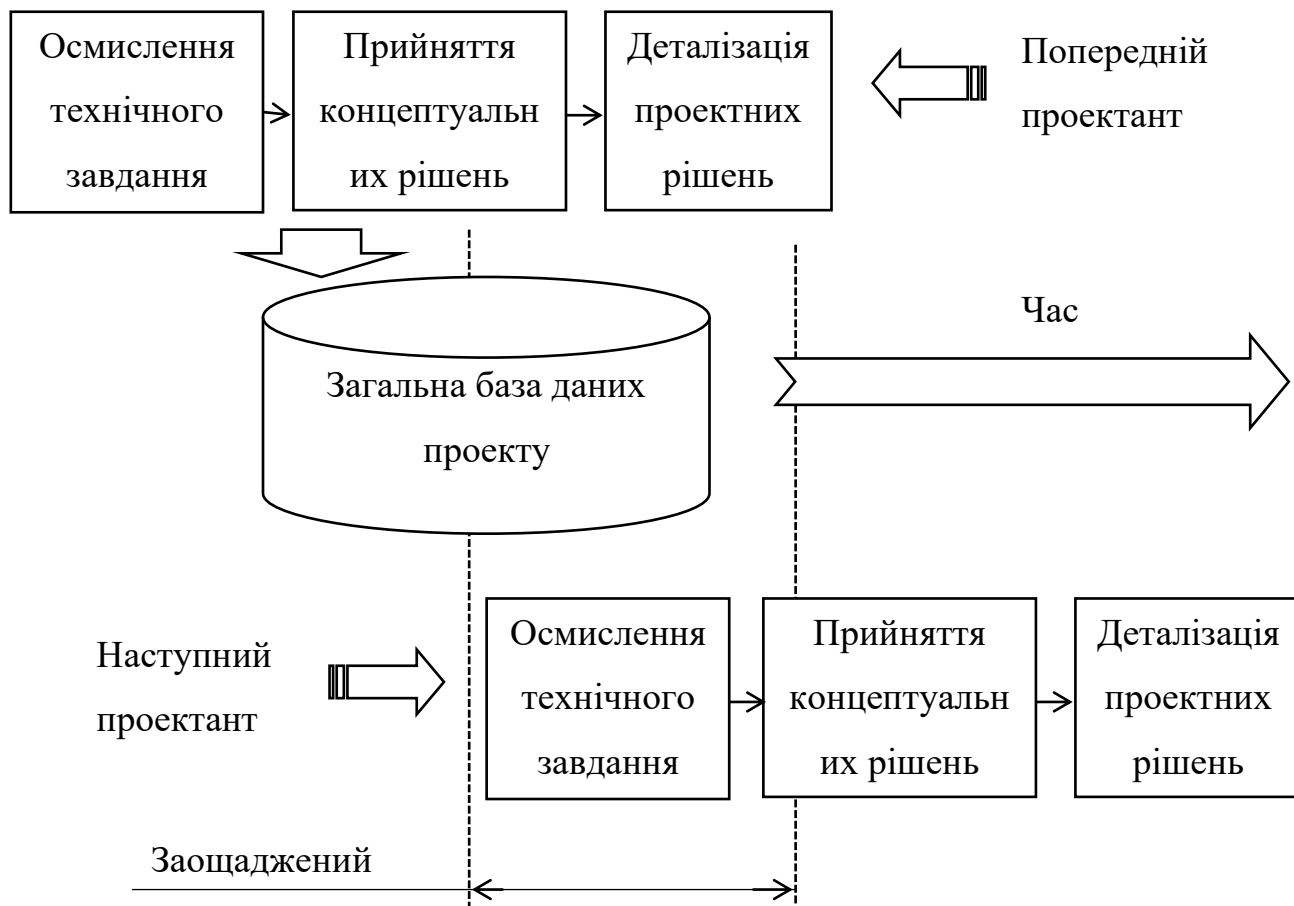


Рисунок 1.5 – Схема використання паралельного проєктування

Важливе завдання, розв'язуваної SniarTeam, полягає в організації електронних архівів. Електронний архів є не просто набором сканованих документів або CAD-файлів, створених конструкторами. Для кожного документа проєкту в електронному архіві зберігається відповідна інформація, що описує всі дії, виконані над документом протягом усього його життєвого циклу (зміна, тиражування, видача за заявками тощо).

SniarTeam дає змогу керівникам підрозділів працювати в єдиному інформаційному середовищі разом зі своїми фахівцями. Для цього розроблені спеціальні функції, зокрема, RedLining (використання "червоного олівця" для внесення зауважень під час перевірки результатів діяльності своїх підлеглих); засоби WorkFlow, на основі яких керівники можуть контролювати і керувати потоками виробничих завдань. До того ж, у розпорядженні керівника є всі

можливості з пошуку та перегляду інформації щодо проєктів. Швидкі отримання відповідей на запитання: "Які документи необхідно зробити до зазначеної дати?", "Які документи мали бути зроблені до зазначеної дати, але не виконані?", "Де знаходиться даний документ?" тощо, дозволяють своєчасно і правильно приймати рішення щодо планування робіт і керування підрозділами.

Інформація в SmarTeam організована у вигляді проєктів. Проєкт подається як ієрархічне дерево, що описує зв'язки між вхідними в проєкт об'єктами, а саме: дерево проєкту виробу описує зв'язки між деталями і складовими одиницями виробів (рисунок 1.6).

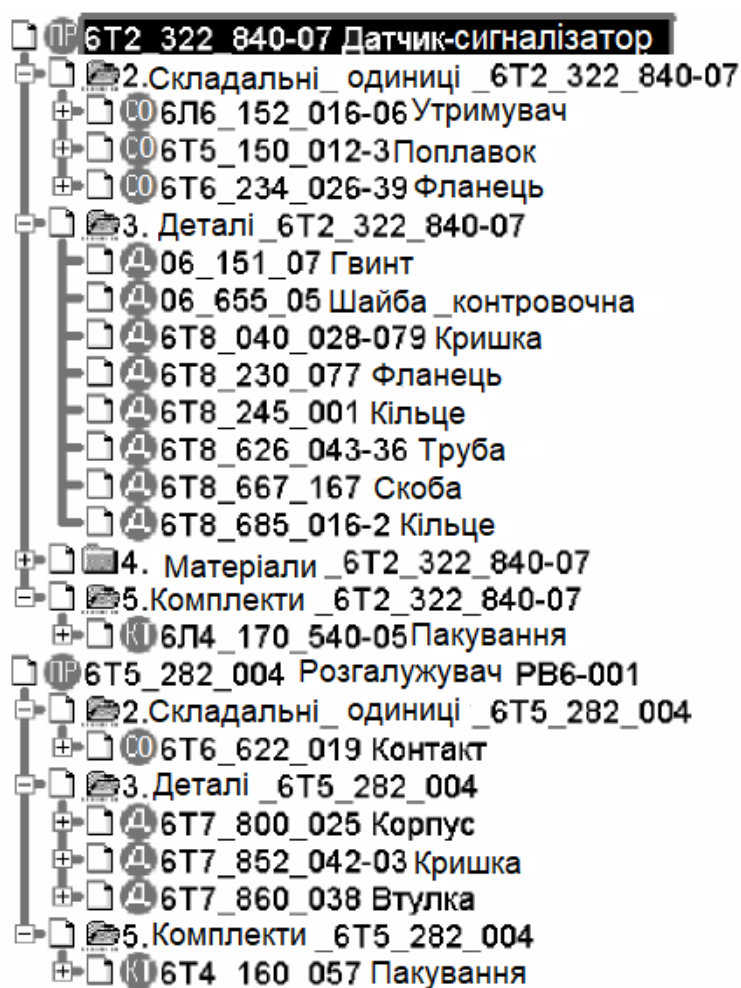


Рисунок 1.6 – Дерево проєкту в PDM SmarTeam

Кожен проект характеризується обліковою карткою, що заповнюється на початку створення проекту. В ній зберігається номер проекту, його найменування, прізвище керівника, дата створення тощо.

Для входу в проект потрібно вибрати його зі списку проектів. При цьому в лівій частині екрана з'явиться дерево проекту, а в правій – облікова картка об'єкта, обраного на даний момент. (рисунок 1.7). У полях облікової картки об'єкта зафіксовані його позначення, найменування, дата створення, прізвище розробника тощо. До того ж, вказується тип, ім'я та розташування файлу, що містить об'єкт.

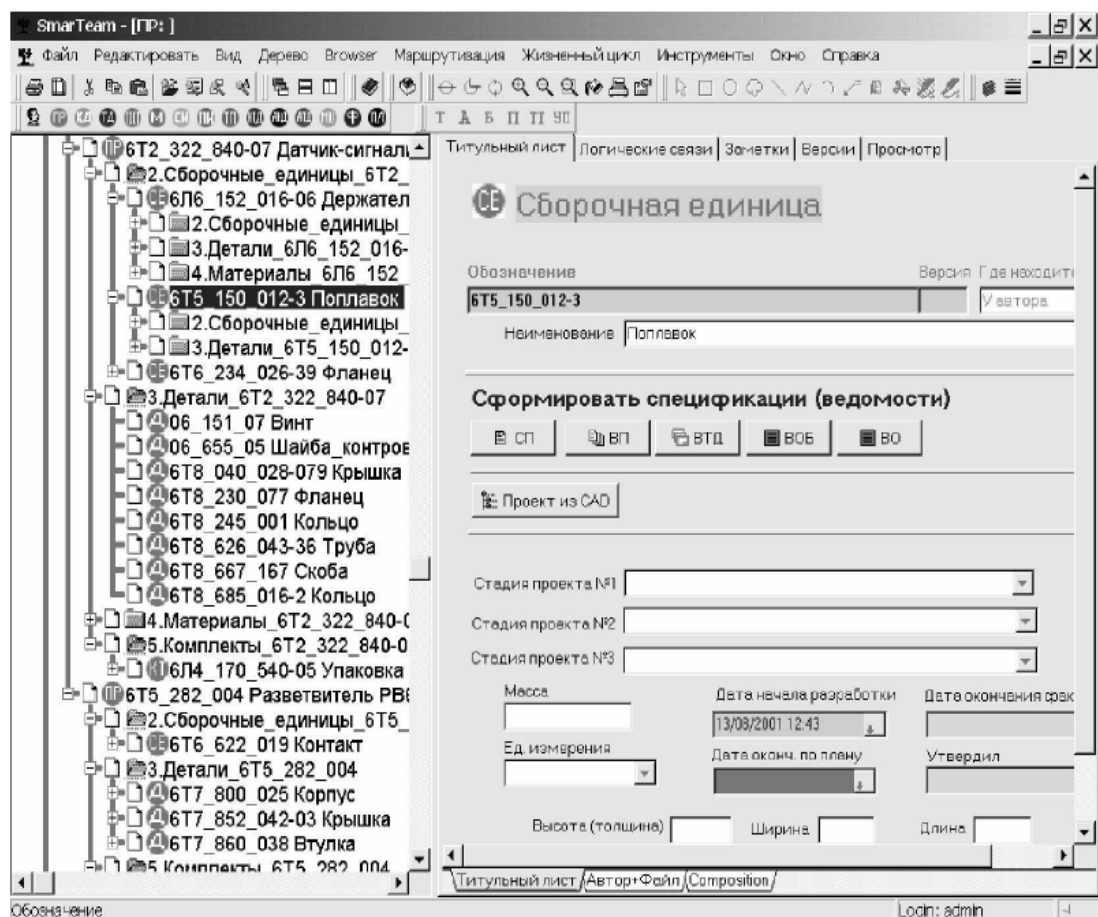


Рисунок 1.7 – Облікова картка об'єкта проектування

У додаткових розділах облікової картки об'єкта зазначені його логічні зв'язки з іншими об'єктами в дереві проекту (розділ "Зв'язки"), а також вказується стан об'єкта в поточний момент (розділ "Зміни"). За розділом

"Зміни" можна простежити всі етапи проходження об'єкта (документа) в процесі проєктування.

Функції SmarTeam дають можливість будувати та редагувати дерево проєкту, створювати нові об'єкти, заповнювати та редагувати облікові картки тощо. Для того, щоб перейти до роботи з файлом об'єкта, досить вказати цей об'єкт у дереві проєкту і вибрати команду "Редагувати". SmarTeam за типом файлу автоматично визначить потрібний додаток та активізує його з одночасним відкриттям даного файлу.

Інша команда "Перегляд" дає можливість побачити графічне зображення об'єкта, не входячи до режиму редагування (рисунок 1.8). Таке зображення подано як слайд, який був автоматично збережений під час попередньої роботи з об'єктом.

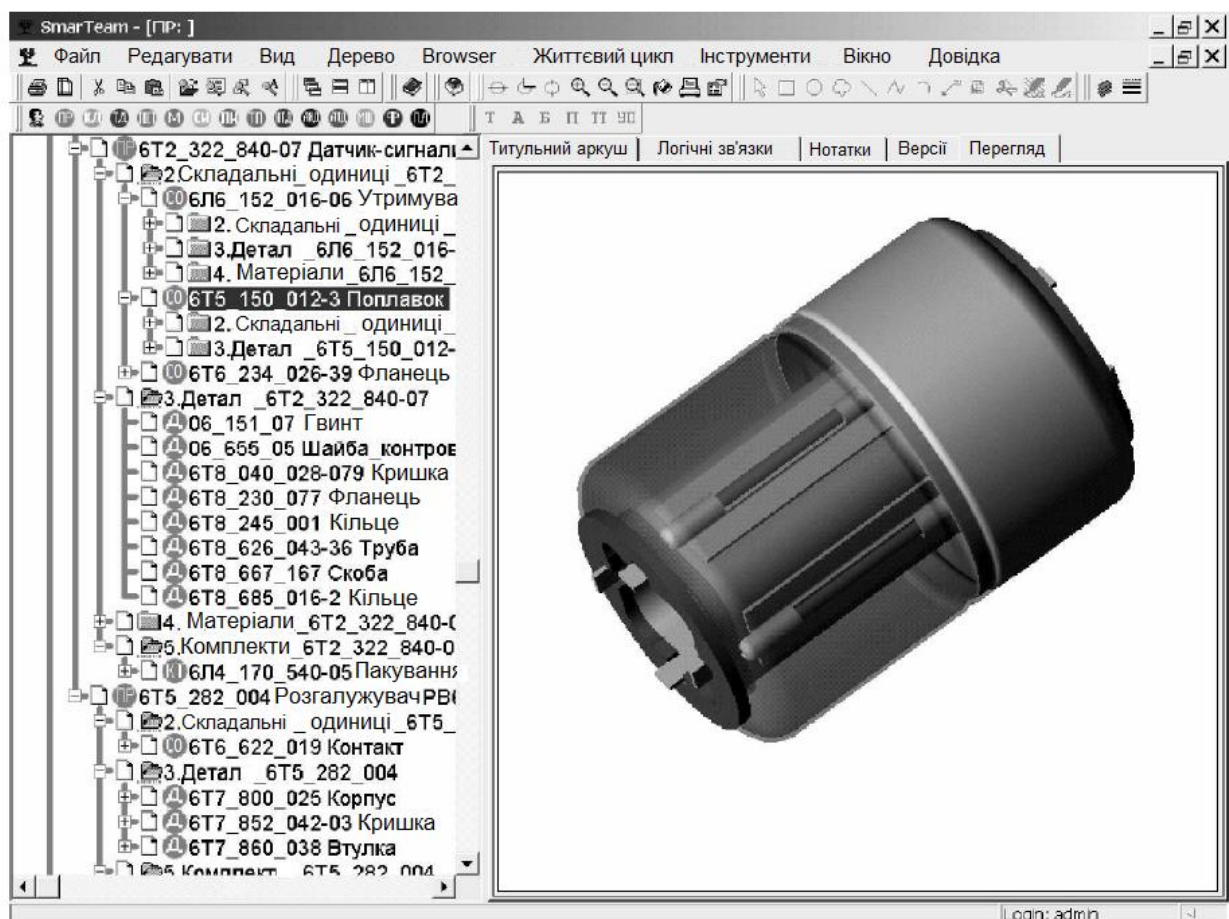


Рисунок 1.8 – Графічний огляд об'єкта проєктування

Користувач може здійснювати автоматичний пошук необхідної інформації в проєкті. Для цього необхідно задати до системи відповідний запит (критерій пошуку). Можна використовувати як готові запити, так і створювати нові.

Єдина база даних і засоби налаштування SmarTeam дозволяють організувати роботу технологів, конструкторів та інших фахівців підприємства в єдиному інформаційному просторі (рисунок 1.9).

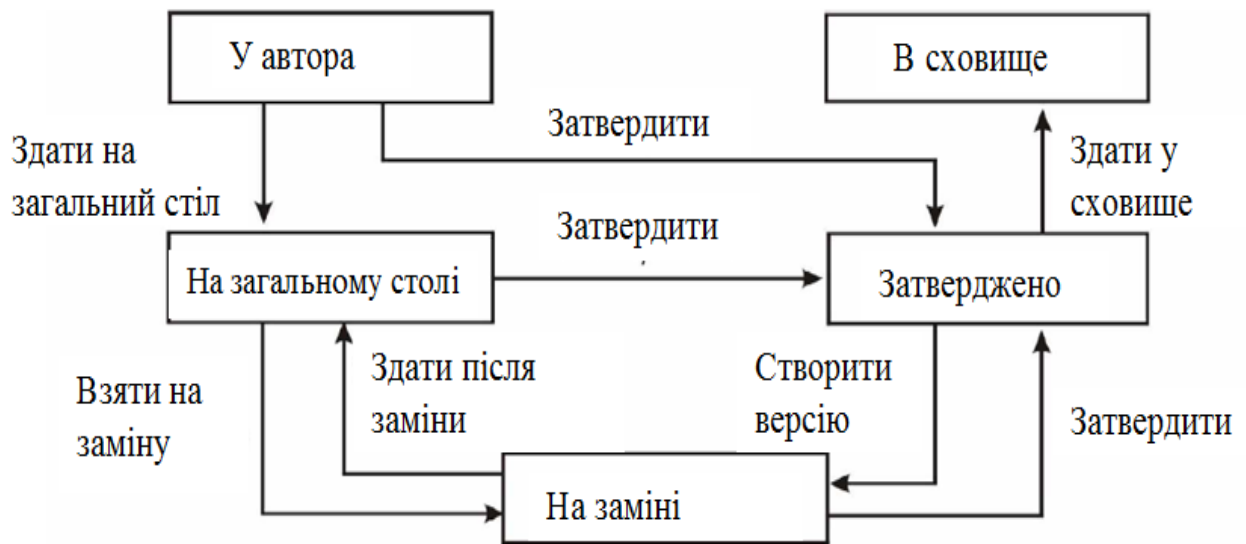


Рисунок 1.9 – Стан об'єкта (документа)

Це дозволяє конструкторам та працівникам цеху чи нормувальникам витрат матеріалів, а також іншим фахівцям запозичувати і використовувати запроваджену раніше технологіями (під час проєктування технологічних процесів) інформацію про виконувані операції, необхідний інструмент, устаткування тощо.

2 МОДЕЛЮВАННЯ РОБОТИ ПРОГРАМНОГО МОДУЛЯ

2.1 Розробка архітектури програмного модуля

У межах багаторівневого уявлення в програмних системах можна виділити три групи функцій, які орієнтовані на розв'язання різних підзадач:

- функції введення і відображення даних (забезпечують взаємодію з користувачем);
- прикладні функції, що характерні для даної предметної області;
- функції керування ресурсами (файловою системою, базою даних тощо).

Виконання цих функцій в основному забезпечується програмними засобами, які можна представити як взаємопов'язані компоненти, в яких:

- графічний компонент відповідає за користувацький інтерфейс;
- прикладний компонент реалізує алгоритм розв'язання конкретного завдання;
- компонент керування ресурсом забезпечує доступ до необхідних ресурсів.

Архітектура «клієнт–сервер» окреслює загальні принципи організації взаємодії в мережі, в яких є сервери, вузли–постачальники деяких специфічних функцій (сервісів) і клієнти, споживачі цих функцій.

У будь–якій мережі, що побудована на сучасних мережевих технологіях, присутні елементи взаємодії клієнт–сервер, найчастіше на основі дворівневої архітектури. Дворівневою називають її через необхідність розподілу трьох базових компонентів між двома вузлами (клієнтом і сервером).

Дворівнева архітектура застосовується в клієнт–серверних системах, в яких сервер відповідає на клієнтські запити безпосередньо і в повному обсязі, при цьому застосовуючи лише власні ресурси. Тобто сервер не використовує сторонні мережеві додатки і, відповідно, не звертається до сторонніх ресурсів для виконання будь–якої частини запиту.

Архітектуру програмної системи, що розроблюється, наведено на рисунку 2.1.

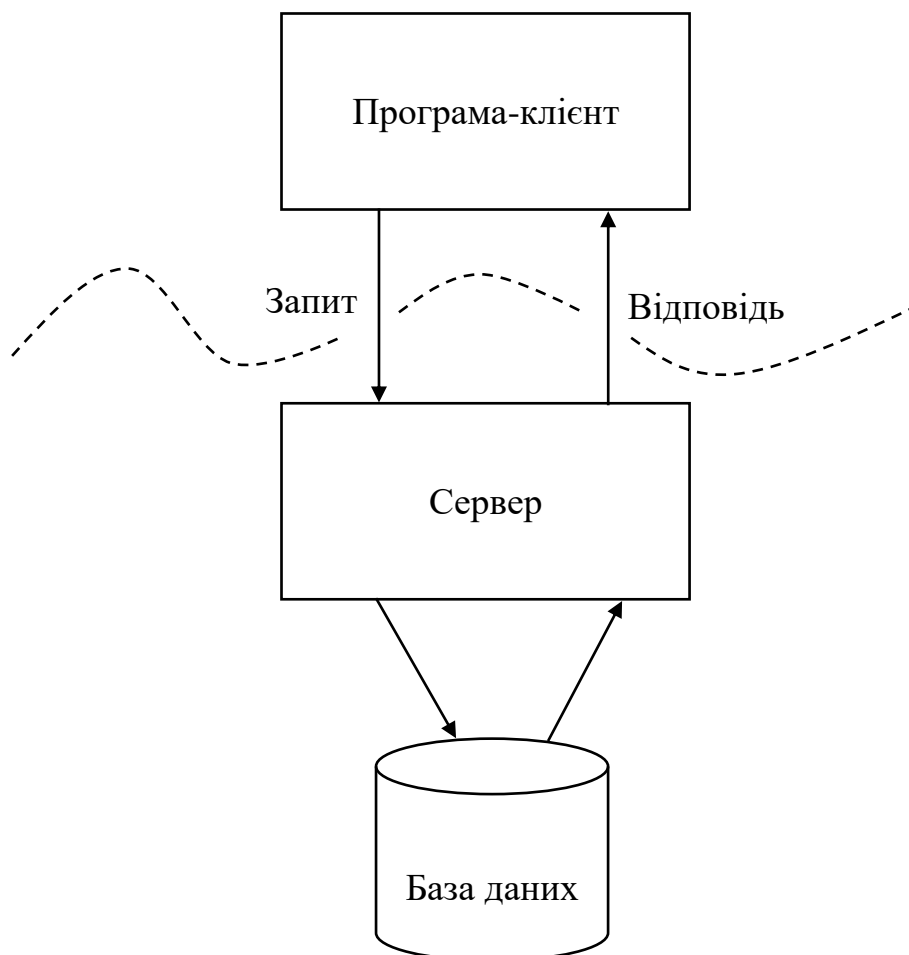


Рисунок 2.1 – Архітектура програмного модуля

Дана архітектура передбачає, що графічну оболонку з набором необхідних функцій (клієнтський процес) завантажують на одному чи декількох комп'ютерах, а процес-сервер – на іншому. За приклад сервера використаємо окремий комп'ютер з операційною системою Linux та системою керування базами даних PostgreSQL.

Для обміну інформацією між компонентами архітектури застосуємо мережу Internet.

У такій моделі один процес може працювати незалежно від іншого, а також виконувати певні завдання та розділяти обчислювальне навантаження.

Кожен з клієнтів може одночасно обробляти різні задачі чи виконувати різні функції. Наприклад, на першому ПК можна завантажити функцію формування специфікації, а на другому оператор може наповнювати довідники новим деталями, вже на третьому – створювати технології виготовлення деталей.

Загальну архітектуру взаємодії між компонентами програмної системи представлено на рисунку 2.2.



Рисунок 2.2 – Загальна архітектура взаємодії між компонентами програмного модуля

2.2 Моделювання поведінки програмного модуля

2.2.1 Розробка діаграми варіантів використання програмної системи

Для моделювання поведінки програмної системи звернемося до діаграми UML.

Розробка діаграми варіантів використання окреслює такі цілі:

- визначення спільних кордонів та контексту предметної області, що моделюється на початкових етапах проектування системи;
- формулювання загальних вимог до функціональної поведінки системи;
- розроблення вихідної концептуальної моделі системи для її подальшої деталізації у формі логічних і фізичних моделей;
- підготування вихідної документації для взаємодії розробників системи з її замовниками та користувачами.

Призначення даної діаграми полягає в наступному: система, що проектується, подається як сутності або актори, що взаємодіють з системою за допомогою, так званих, варіантів використання. Проте актором або дійовою особою може бути будь-яка сутність, яка взаємодіє з системою ззовні, наприклад, людина, технічний пристрій, програма або будь-яка інша система, що може слугувати джерелом впливу на модельовану систему так, як визначить сам розробник.

Варіант застосування використовують для опису сервісів, які система надає акторові. Таким чином, кожен варіант використання визначає певний набір дій, який є характерним при діалозі системи з актором.

Діаграму варіантів використання нашої програмної системи в режимі створення довідників зображено на рисунку 2.3.

Як можна бачити на рисунку 2.3 під час створення довідників за допомогою програми, що розробляється, оператор може застосовувати такі функції програми, як:

- створення довідника товарно-матеріальних цінностей (ТМЦ);
- створення довідника конструкторсько-технологічної документації (КТД);
- створення технології виготовлення деталі;
- створення специфікації.

Ми визначили основні функції, але додатково існує багато допоміжних, які не представлені на даній діаграмі:

- створення довідника форматів;
- створення довідника пристроїв;

- створення довідника технологічних операцій;
- створення довідника розмірів;
- створення довідника одиниць вимірювання ;
- створення довідника розрядів робітників;
- створення довідника відділів;
- створення довідника працівників.

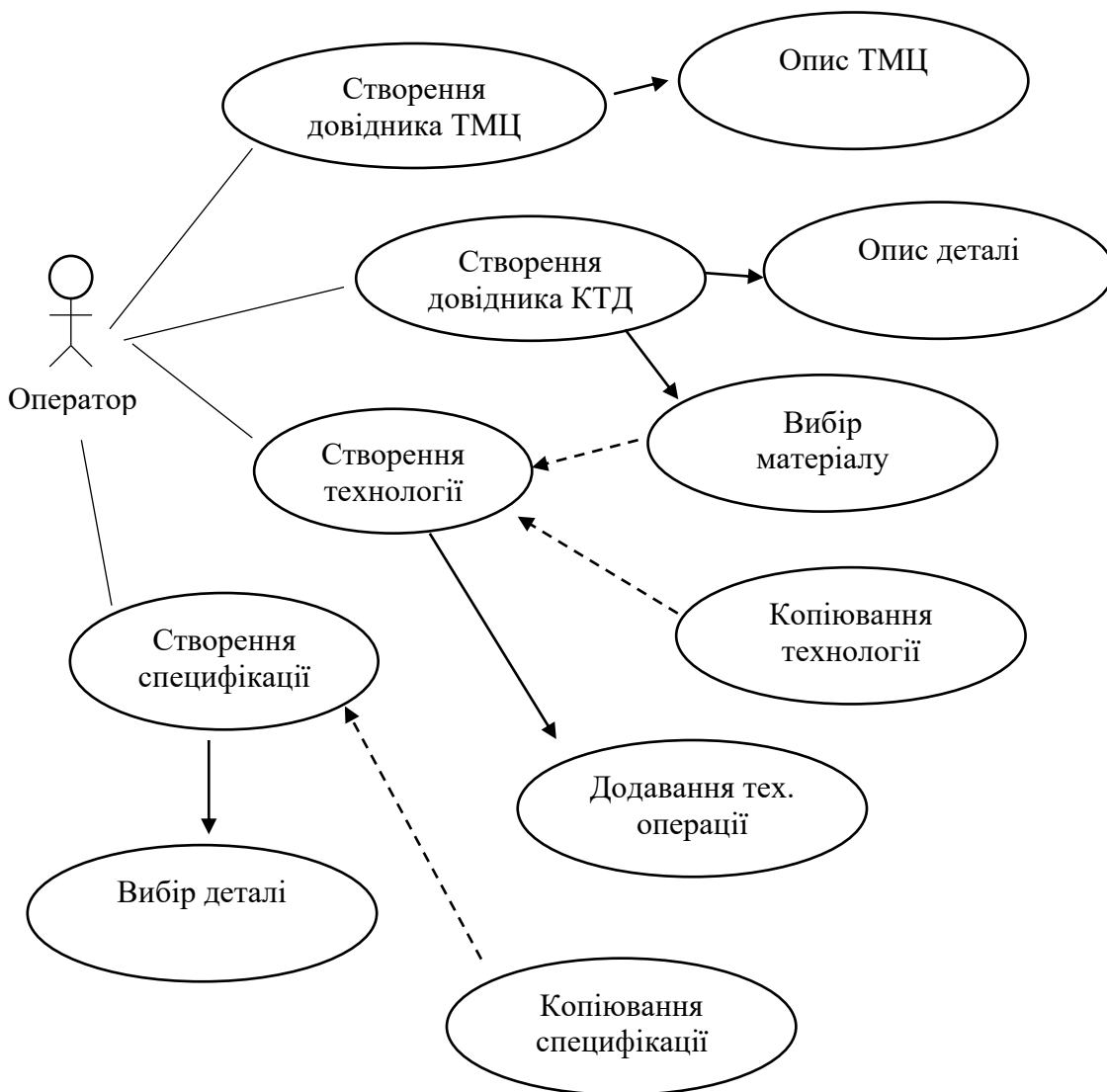


Рисунок 2.3 – Діаграма варіантів використання програми, що розробляється, в режимі створення довідників

Дані функції не було відображено на рисунку 2.3, щоб не перекривати основні варіанти використання.

Як можна бачити на рисунку 2.3 під час створення довідника ТМЦ треба вказати всі характеристики деталі, зокрема:

- коротку назву ТМЦ;
- повну назву ТМЦ;
- технологічні одиниці вимірювання;
- коефіцієнт перерахунку до економічних одиниць вимірювання;
- економічні одиниці вимірювання;
- допуск на порізку;
- торгову марку;
- обліковий рахунок;
- додаткові економічні одиниці вимірювання.

Крім того слід вказати й додаткові дані:

- до якої категорії обліку належать ТМЦ (матеріал, стандартні вироби, інші вироби, інструмент, готова продукція);
- технічну розмітність.

Під час створення КТД необхідно вказати такі дані:

- децимальний номер деталі;
- найменування деталі;
- матеріал, з якого зроблено деталь;
- первинне застосування (в якій складальній одиниці вперше застосовано деталь);
- формат креслення деталі;
- вагу деталі.

Під час використання функції «Створення технології» програма дозволяє:

- створювати базу технологію виготовлення;
- копіювати технологію, розроблену раніше;
- змінювати чи обирати матеріал деталі;
- додавати технологічну операцію до переліку операцій щодо виготовлення деталі.

Створення специфікації виконується тільки для складальної одиниці та містить такі функції:

- вибір та додавання до специфікації матеріалів, деталей, документів, стандартних виробів, інших складальних одиниць;
- копіювання специфікації.

2.2.2 Розробка діаграми станів

Для моделювання поведження програмної системи на логічному рівні в мові UML можна застосовувати відразу кілька канонічних діаграм: станів, діяльності, послідовності й кооперації, кожна з них фіксує увагу на окремому аспекті функціонування системи.

Особливість діаграми станів, на відміну від інших діаграм, полягає в тому, що вона описує процес зміни станів тільки одного класу, а точніше – одного екземпляра певного класу, іншими словами, моделює всі можливі зміни в стані конкретного об'єкта. Проте зміна стану об'єкта може бути викликана зовнішніми впливами з боку інших об'єктів або ззовні. Власне для опису реакції об'єкта на подібні зовнішні впливи й застосовують діаграми станів.

Діаграму станів для моделювання роботи програми наведено на рисунку 2.4.

На рисунку 2.4 наочно подано, що для того, щоб розробити технологію виготовлення деталі необхідно пройти такі стадії:

- створення допоміжних довідників;
- створення довідника ТМЦ;
- створення довідника КТД;
- створення довідника специфікації;
- створення технології.

Перехід з одного етапу на інший можливо реалізувати тільки після завершення всіх встановлених операцій на кожному окремому етапі.

Опрацюємо порядок проходження всіх етапів.

На першому етапі необхідно створити всі допоміжні довідники.

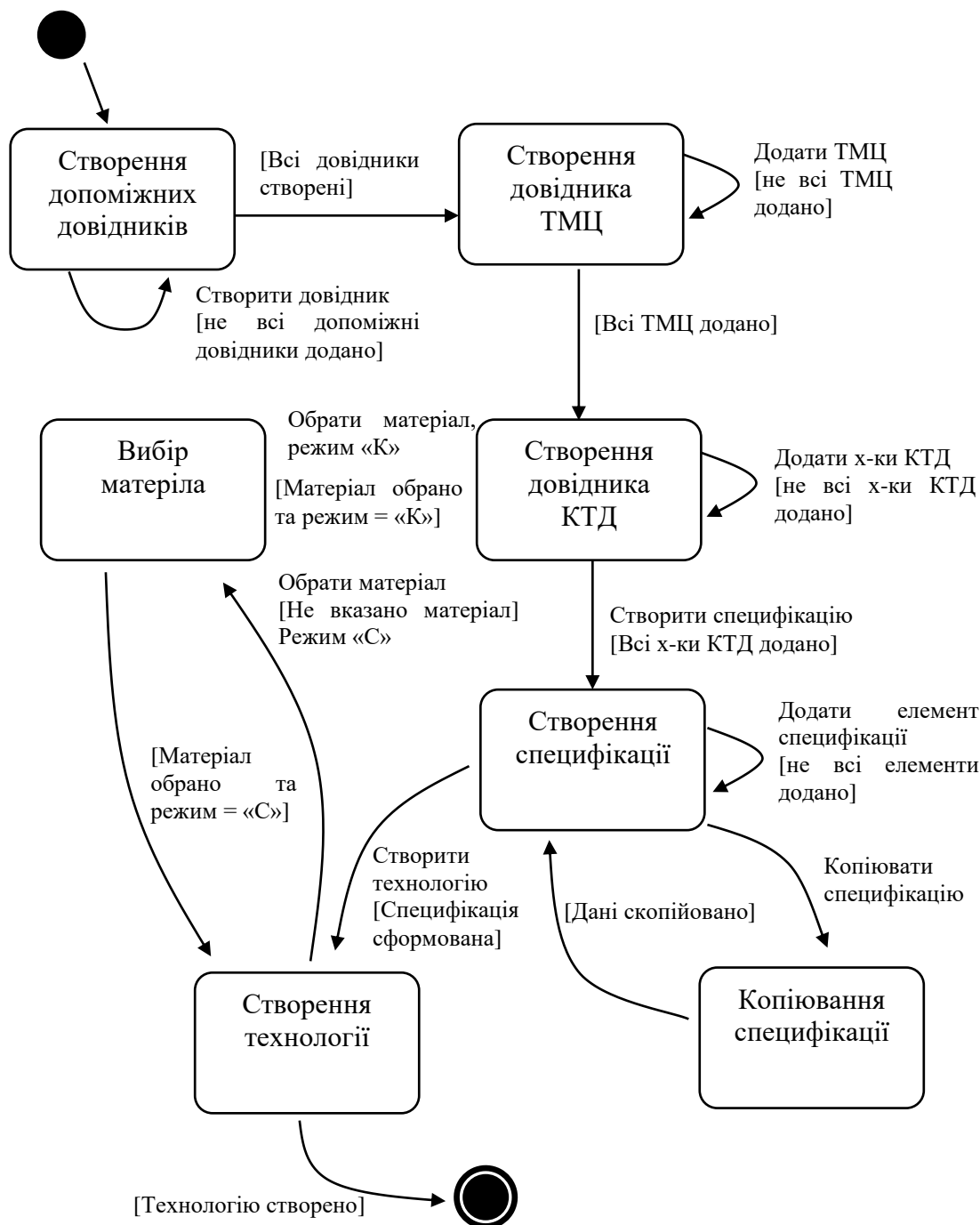


Рисунок 2.4 – Діаграма станів для моделювання роботи програми

Тільки коли всі вони будуть готові, можна переходити до процесу створення довідника ТМЦ. Як було зазначено вище, до довідника ТМЦ треба додавати повний опис кожного матеріалу.

Закінчивши опис ТМЦ, можна переходити до створення довідника КТД. На цьому етапі необхідно повністю заповнити всі характеристики деталі, або складальної одиниці. Якщо конструктор ще не зміг вибрати матеріал для

виготовлення деталі, то його можна не вказувати, але без даної інформації не можливо буде створити технологію виготовлення цієї деталі.

Потім, коли всі характеристики деталі будуть вказані, можна переходити до процесу зі створення специфікації. На цьому етапі можна скопіювати створену раніше специфікацію чи додати всі позиції самостійно.

Під час додавання всіх позицій можна переходити до створення технології. На цьому етапі треба зазначити всі операції виготовлення, а також задати необхідні коефіцієнти та параметри. Крім того, для завершення створення технології треба обов'язково вказати матеріал виготовлення деталі.

Після завершення всіх вказаних стадій, можна вважати, що технологію створено.

3 РОЗРОБКА СТРУКТУРИ БАЗИ ДАНИХ

3.1 Обирання системи керування базами даних

В архітектурі розробляємої програми основним компонентом є система керування базою даних (СКБД).

У даному разі як зразок такої системи будемо використовувати PostgreSQL – широко розповсюджену систему керування базами даних з відкритим сирцевим кодом.

На сучасному етапі PostgreSQL використовується по всьому світі для реалізації великих систем, зокрема, системи аналізу фінансових даних, пакету моніторингу функціональності потоків, як система обліку медичної інформації, СКБД географічних систем.

Крім того, PostgreSQL використовувався в багатьох світових університетах як навчальний інструмент.

Система керування базами даних PostgreSQL є сукупністю програмних і лінгвістичних засобів загального або спеціального призначення, що забезпечують керування створенням та використанням баз даних.

Основні функції СКБД PostgreSQL:

- керування даними, що розташовані в зовнішній пам'яті (на дисках);
- керування даними, що розташовані в оперативній пам'яті з використанням дискового кеша;
- запис змін до журналу, резервне копіювання, а також відновлення бази даних після збоїв;
- підтримка мов БД (мова визначення даних, мова маніпулювання даними).

СКБД PostgreSQL складається з таких компонентів:

- ядра, що відповідає за керування даними як у зовнішній, так й оперативній пам'яті та журналізацію;

- процесора мови БД, який забезпечує оптимізацію запитів на вилучення та зміну даних і створення, зазвичай, машинно–незалежного виконуваного внутрішнього коду;

- підсистеми підтримки часу виконання, що інтерпретує програми маніпуляції даними, створює користувацький інтерфейс із СКБД;

- сервісних програм (зовнішніх утиліт), які забезпечують низку додаткових можливостей щодо обслуговування інформаційної системи.

Звернемо увагу на основні можливості СКБД PostgreSQL.

Функціями PostgreSQL є блоки коду. Вони виконуються на сервері, а не клієнті БД, хоча можуть бути написані на чистому SQL. Реалізація додаткової логіки, на кшталт, умовних переходів і циклів, виходить за рамки власне SQL, тому вимагає використання деяких мовних розширень. Функції можна писати, використовуючи одну з нижченаведених мов:

- вбудовану процедурну мову PL/pgSQL, яка багато в чому аналогічна до мови PL/SQL, яка використовується в СКБД Oracle;

- скриптову мову – PL/Lua, PL/LOLCODE, PL/Perl, plPHP, PL/Python, PL/Ruby, PL/sh, PL/Tcl і PL/Scheme;

- класичні мови – C, C++, Java (через модуль PL/Java);

- статистичні мови R (через модуль PL/R).

PostgreSQL дозволяє використання функцій, які повертають набір записів, використовувати який можна так само, як і результат виконання звичайного запиту.

Функції можуть реалізовуватись як з правами їхнього автора, так і з правами поточного користувача.

Іноді функції ототожнюються зі збереженими процедурами, проте між цими поняттями різниця є.

Тригери визначаються як функції, що ініційовані DML–операціями. Приміром, операція INSERT може активізувати тригер, який перевіряє доданий запис на відповідності до певних умов. Під час написання функцій для тригерів можна послуговуватись різними мовами програмування.

Тригери асоціюються з таблицями. Множинні тригери виконуються за

алфавітним порядком.

Механізм правил – це механізм створення користувацьких обробників не тільки DML–операцій, але й операції вибірки. Основною відмінністю від механізму тригерів є те, що правила спрацьовують на етапі розбору запиту, до вибору оптимального плану виконання і самого процесу виконання. Правила дозволяють змінити поведінку системи під час виконання SQL–операції до таблиці.

Актуальним прикладом є реалізація механізму уявлень: під час створення уявлення діє правило, за яким замість виконання операції вибірки до уявлення система повинна виконувати операцію вибірки до базової таблиці з урахуванням умов вибірки, що покладені в основі визначення уявлення. Для створення уявлень, які підтримують операції оновлення правила для операцій вставки, зміни і видалення рядків повинні визначатися користувачем.

У PostgreSQL наявна підтримка індексів таких типів: B–дерево, хеш, R–дерево, GiST, GIN(4). За необхідності можна створювати нові типи індексів, хоча це далеко не тривіальний процес. Індокси в PostgreSQL наділені такими властивостями:

- можливий перегляд індексу як у прямому, так й зворотному порядку: створювати окремий індекс для роботи конструкції ORDER BY ... DESC не потрібно;

- можливе створення індексу над кількома стовпцями таблиці, а також – над стовпцями різних типів даних;

- індекси можуть бути функціональними, іншими словами, будуватися не на базі набору значень певного стовпця/стовпців, а на базі набору значень функції від набору значень;

- індекси можуть бути частковими, іншими словами, будуватися тільки за частиною таблиці за деякою її проекції);

- в окремих випадках це допомагає створювати більш компактні індекси або досягати покращення продуктивності за рахунок застосування різних типів індексів для різних (наприклад, з точки зору частоти оновлення) частин таблиці;

– планувальник запитів може одночасно використовувати кілька індексів для виконання складних запитів.

3.2 Структури сутності бази даних

Проаналізувавши вихідні дані та моделювання роботи програмної системи, ми встановили першорядний набір даних, який потребує зберігання й та обробки.

Основні об'єкти, характеристики яких потребують опису:

- товарно–матеріальна цінність;
- деталь;
- картка технологічної операції;
- технологія виготовлення деталі;
- специфікація.

Кожна сутність має набір полів, які асоціюються з тим чи іншим об'єктом.

Структуру сутності «Товарно–матеріальна цінність» наведено в таблиці 3.1.

Таблиця 3.1 – Структура сутності «Товарно–матеріальна цінність»

Найменування	Тип даних	Ключ
ID_TMC	Serial	X
ShortName_TMC	Text	
FullName_TMC	Text	
ID_Unit	Integer	
K_EconomicUnit	Text	
ID_EconomicUnit_List	Integer	
Dopusk_Cut	Text	
ID_TradeMark_List	Integer	
ID_Account	Integer	

Продовження таблиці 3.1

Найменування	Тип даних	Ключ
ID_DopEconomicUnit_1_List	Integer	
ID_DopEconomicUnit_2_List	Integer	
K_DopEconomicUnit_1	Text	
K_DopEconomicUnit_2	Text	
Set_Material	Boolean	
Set_StandartProduct	Boolean	
Set_OtherProduct	Boolean	
Set_Tool	Boolean	
Set_CompliteProduct	Boolean	
Type_TMC	Text	
MinKol	Text	
ID_Folder_TMC_List	Integer	
ID_Dop_Material	Integer	

У таблиці 3.1 наведені поля, що призначені для:

- ID_TMC – унікальний код для ідентифікації рядка таблиці;
- ShortName_TMC – коротке найменування ТМЦ;
- FullName_TMC – повне найменування ТМЦ;
- ID_Unit – посилання на таблицю з технологічними одиницями вимірювання;
- K_EconomicUnit – коефіцієнт перетворення технічної одиниці вимірювання на економічну;
- ID_EconomicUnit_List – посилання на таблицю з економічними одиницями вимірювання;
- Dopusk_Cut – допуск на порізання;
- ID_TradeMark_List – посилання на таблицю зі списком торговельних марок;
- ID_Account – посилання на таблицю рахунків;
- ID_DopEconomicUnit_1_List – посилання на таблицю з додатковими

економічними одиницями вимірювання;

- ID_DopEconomicUnit_2_List – посилання на таблицю з додатковими економічними одиницями вимірювання;

- K_DopEconomicUnit_1 – коефіцієнт перетворення технічної одиниці вимірювання на додаткову економічну;

- K_DopEconomicUnit_2 – коефіцієнт перетворення технічної одиниці вимірювання на додаткову економічну;

- Set_Material – ознака того, що дана ТМЦ – матеріал;

- Set_StandartProduct – ознака того, що дана ТМЦ – стандартні вироби;

- Set_OtherProduct – ознака того, що дана ТМЦ – інша продукція;

- Set_Tool – ознака того, що дана ТМЦ – інструмент;

- Set_CompliteProduct – ознака того, що дана ТМЦ – готова продукція;

- Type_TMC – тип ТМЦ;

- MinKol – мінімальна кількість ТМЦ на складі;

- ID_Folder_TMC_List – посилання на папку, до якої належить ТМЦ;

- ID_Dop_Material – посилання на таблицю з допоміжними матеріалами.

Моделювання структури даних було виконано в автоматизованому середовищі PowerDesigner.

PowerDesigner дає змогу моделювати структуру даних на етапі проектування, пропонує техніку керування метаданими і складається з унікальної технології аналізу взаємозв'язків моделей. Водночас із підтримкою провідних технік моделювання та керування метаданими, PowerDesigner також дозволяє працювати з моделями будь-яких типів в єдиному інтегрованому середовищі, а репозиторій метаданих PowerDesigner сприяє налагодженню взаємодії між усіма зацікавленими особами компанії, що забезпечує більш швидкий відгук на зміни в чинному бізнес-середовищі.

Фізичну структуру таблиці «Товарно-матеріальна цінність» показано на рисунку 3.1.

Name	Code	Data Type	Length	Preci	M	P	D	Domain
ID_TMC	ID_TMC	Serial			<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
ShortName_TMC	ShortName_TMC	Text			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
FullName_TMC	FullName_TMC	Text			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
ID_Unit	ID_Unit	Integer			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
K_EconomicUnit	K_EconomicUnit	Text			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
ID_EconomicUnit_List	ID_EconomicUnit_List	Integer			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
Dopusk_Cut	Dopusk_Cut	Text			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
ID_TradeMark_List	ID_TradeMark_List	Integer			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
ID_Account	ID_Account	Integer			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
ID_DopEconomicUnit_1_List	ID_DopEconomicUnit_1_List	Integer			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
ID_DopEconomicUnit_2_List	ID_DopEconomicUnit_2_List	Integer			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
K_DopEconomicUnit_1	K_DopEconomicUnit_1	Text			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
K_DopEconomicUnit_2	K_DopEconomicUnit_2	Text			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
Set_Material	Set_Material	Boolean			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
Set_StandartProduct	Set_StandartProduct	Boolean			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
Set_OtherProduct	Set_OtherProduct	Boolean			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
Set_Tool	Set_Tool	Boolean			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
Set_CompliteProduct	Set_CompliteProduct	Boolean			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
Type_TMC	Type_TMC	Text			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
MinKol	MinKol	Text			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
ID_Folder_TMC_List	ID_Folder_TMC_List	Integer			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
ID_Dop_Material	ID_Dop_Material	Integer			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>

Рисунок 3.1 – Фізична структура таблиці «Товарно–матеріальна цінність»

У таблиці 3.2 подано структуру сутності «Деталь».

Таблиця 3.2 – Структура сутності «Деталь»

Найменування	Тип даних	Ключ
ID_Detail	Serial	X
Name_Detail	Text	
DecimalNumber	Text	
ID_Material_List	Integer	
Count_Detail	Text	
Count_Detail_From	Text	
Weight_Detail	Text	
Notice	Text	
Type	Text	
Visible_Detail	Boolean	
Image_Detail_Path	Text	
Equipment	Boolean	

Продовження таблиці 3.2

Найменування	Тип даних	Ключ
ID_First_Assemble	Integer	
ID_Format	Integer	
ID_Folder_Detail	Integer	

У таблиці 3.2 наведено поля, що призначені для:

- ID_Detail – унікальний код для ідентифікації рядка таблиці;
 - Name_Detail – найменування деталі;
 - DecimalNumber – децимальний номер деталі;
 - ID_Material_List – посилання на таблицю, в якій зберігаються дані про ТМЦ;
 - Weight_Detail – вага деталі;
 - Notice – примітка;
 - Type – ознака типу деталі (звичайна деталь або складальна одиниця);
 - Visible_Detail – ознака, що використовується за умови відображення деталі в дереві КТД;
 - Image_Detail_Path – шлях до зображення деталі;
 - Equipment – ознака, що вказує на те, чи є деталь технологічною оснасткою, чи ні;
 - ID_First_Assemble – посилання на складальну одиницю, що є першим застосуванням деталі;
 - ID_Format – посилання на таблицю, в якій зберігається назва форматів креслень;
 - ID_Folder_Detail – посилання на теку, до якої входить дана деталь.
- Фізичну структуру таблиці «Деталь» подано на рисунку 3.2.

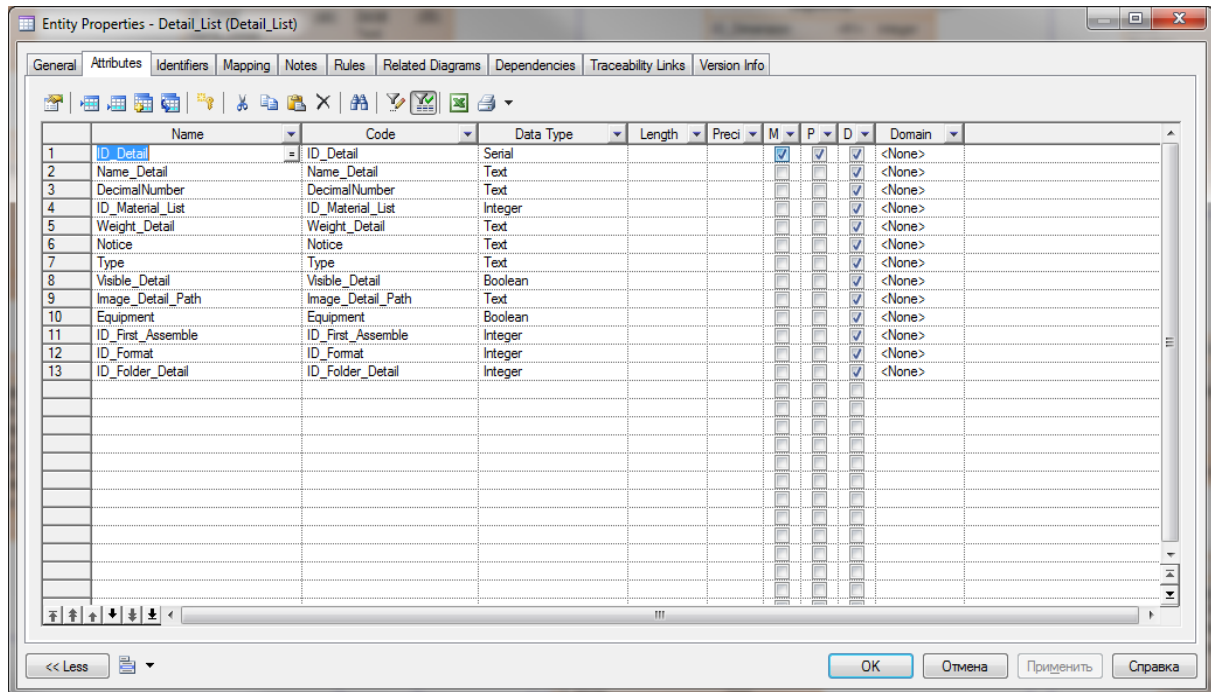


Рисунок 3.2 – Фізична структура таблиці «Деталь»

Структуру сутності «Картка технологічної операції» наведено в таблиці 3.3.

Таблиця 3.3 – Структура сутності «Картка технологічної операції»

Найменування	Тип даних	Ключ
ID_OperationCard	Serial	X
ID_Operation	Integer	
Num_Operation	Integer	
ID_Departments	Integer	
OperationDescribe	Text	
EskizPath	Text	
ID_Device	Integer	
ID_Rank	Integer	
Wokers_Count	Integer	
Part	Boolean	
Norma_Unit	Text	
Trpz_Norm	Text	

Продовження таблиці 3.3

Найменування	Тип даних	Ключ
T_Norm	Text	
ID_Detail	Integer	
Notice	Text	
ID_Technology	Integer	
ID_Files	Integer	

У таблиці 3.3 наведені поля, що призначені для:

- ID_OperationCard – унікальний код для ідентифікації рядка таблиці;
- ID_Operation – посилання на таблицю, в якій зберігається назва технологічних операцій;
- Num_Operation – номер операції за порядком її слідування;
- ID_Departments – посилання на таблицю з переліком підрозділів;
- OperationDescribe – опис технологічної операції;
- EskizPath – шлях до місця зберігання ескізу деталі;
- ID_Device – посилання на таблицю з переліком пристроїв, які необхідні для виготовлення деталі;
- ID_Rank – посилання на таблицю з переліком розрядів робітників;
- Wokers_Count – кількість робітників, яка необхідна для виконання даної операції;
- Part – ознака виготовлення партії деталей;
- Norma_Unit – одиниця нормування;
- Trpz_Norm – підготовчий–заключний час виконання технологічної операції ($T_{пз}$). Його передбачають для ознайомлення з робочими кресленнями, технологічним процесом, а також показника наладки верстата;
- T_Norm – штучний час виконання технологічної операції ($T_{шт}$)
- ID_Detail – посилання на таблицю з переліком деталей;
- Notice – примітка;
- ID_Technology – посилання на таблицю з загальними даними щодо технології виготовлення деталі;

– ID_Files – посилання на таблицю, в якій зберігаються дані, отримані з креслення деталі.

Фізичну структуру таблиці «Картка технологічної операції» наведено на рисунку 3.3.

Name	Code	Data Type	Length	Preci	M	P	D	Domain
ID_OperationCard	ID_OperationCard	Serial			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
ID_Operation	ID_Operation	Integer			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
Num_Operation	Num_Operation	Integer			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
ID_Departments	ID_Departments	Integer			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
OperationDescribe	OperationDescribe	Text			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
EskizPath	EskizPath	Text			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
ID_Device	ID_Device	Integer			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
ID_Rank	ID_Rank	Integer			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
Wokers_Count	Wokers_Count	Integer			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
Part	Part	Boolean			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
Noma_Unit	Noma_Unit	Text			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
Tpz_Norm	Tpz_Norm	Text			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
T_Norm	T_Norm	Text			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
ID_Detail	ID_Detail	Integer			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
Notice	Notice	Text			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
ID_Technology	ID_Technology	Integer			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
ID_Files	ID_Files	Integer			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>

Рисунок 3.3 – Фізична структура таблиці «Картка технологічної операції»

У таблиці 3.4 подано структуру сутності «Технологія».

Таблиця 3.4 – Структура сутності «Технологія»

Найменування	Тип даних	Ключ
ID_Technology	Serial	X
Name_Zakaz	Text	
ID_Detail	Integer	
Kolvo_Detail	Integer	
Count_Detail_From	Text	
Weight_Detail	Text	
Weight_Zagotovka	Text	

Продовження таблиці 3.4

Найменування	Тип даних	Ключ
Complete	Text	
ID_TechOsnastka	Integer	
Notice	Text	
ID_Personal	Integer	
Date_Create	Date	

У таблиці 3.4 наведені поля, що призначені для:

- ID_Technology – унікальний код для ідентифікації рядка таблиці;
- Name_Zakaz – найменування замовлення;
- ID_Detail – посилання на таблицю, в якій зберігаються дані про деталь;
- Kolvo_Detail – кількість деталей, яку треба виготовити у замовленні;
- Count_Detail_From – кількість деталей, яку можна виготовити з однієї заготівлі;
- Weight_Detail – вага деталі;
- Weight_Zagotovka – вага заготівлі;
- Complete – ознака ступеня завершеності замовлення;
- ID_TechOsnastka – посилання на таблицю, в якій зберігаються дані про технологічні оснастки;
- Notice – примітка;
- ID_Personal – посилання на співробітника, що розробив технологію;
- Date_Create – дата створення технології.

Фізичну структуру таблиці «Технологія» наведено на рисунку 3.4.

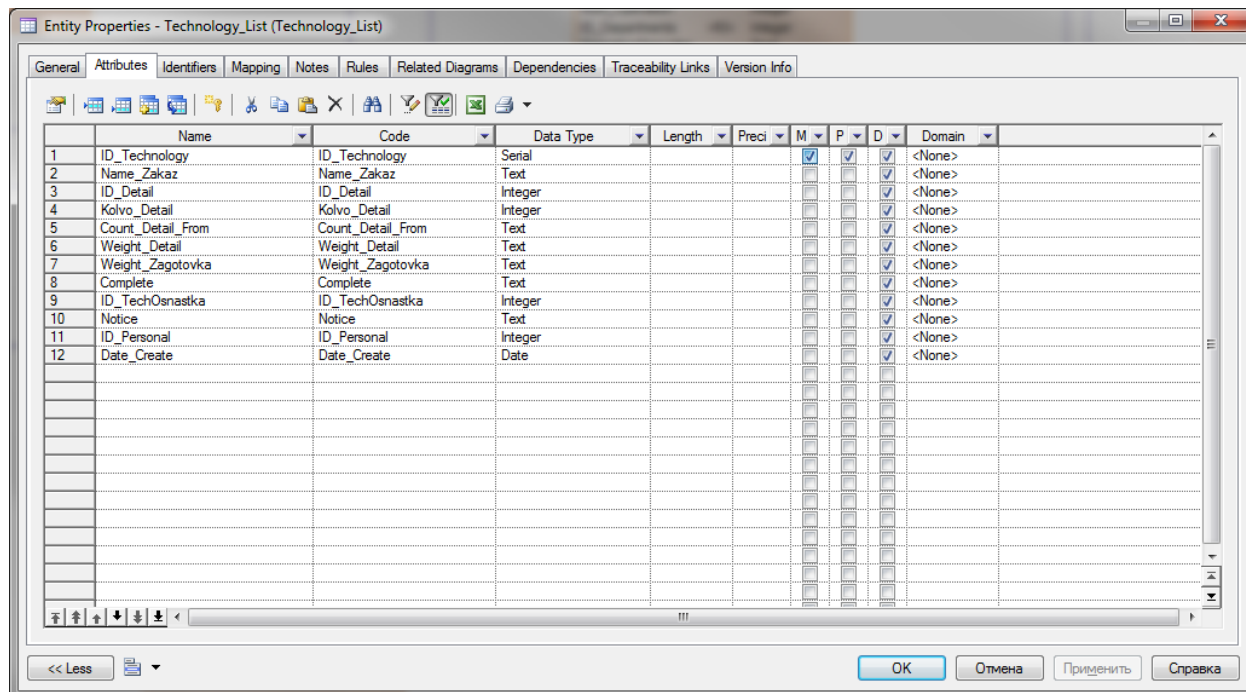


Рисунок 3.4 – Фізична структура таблиці «Технологія»

У таблиці 3.5 наведено структуру сутності «Специфікація».

Таблиця 3.5 – Структура сутності «Специфікація»

Найменування	Тип даних	Ключ
ID_Срес	Serial	X
Path_Document	Text	
ID_Detail	Integer	
ID_Assemble	Integer	
Zone	Text	
Position	Text	
Kolvo	Text	
Notice	Text	
Type	Text	
ID_Format	Integer	

У таблиці 3.5 наведені поля, що призначені для:

- ID_Срес – унікальний код для ідентифікації рядка таблиці;
- Path_Document – шлях до місця зберігання ескізу деталі;

- ID_Detail – посилання на таблицю, в якій зберігаються дані про деталь;
- ID_Assemble – посилання на таблицю, в якій зберігаються дані про складальну одиницю;
- Zone – дані для заповнення поля специфікації «Зона»;
- Position – дані для заповнення поля специфікації «Позиція»;
- Kolvo – дані для заповнення поля специфікації «Кількість»;
- Notice – примітка;
- Type – тип деталі;
- ID_Format – посилання на таблицю, в якій зберігаються дані про формати креслення.

Фізичну структуру таблиці «Специфікація» наведено на рисунку 3.5.

	Name	Code	Data Type	Length	Preci	M	P	D	Domain
1	ID_Спец	ID_Спец	Serial			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
2	Path_Document	Path_Document	Text			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
3	ID_Detail	ID_Detail	Integer			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
4	ID_Assemble	ID_Assemble	Integer			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
5	Zone	Zone	Text			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
6	Position	Position	Text			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
7	Kolvo	Kolvo	Text			<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
8	Notice	Notice	Text			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
9	Type	Type	Text			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>
10	ID_Fomat	ID_Fomat	Integer			<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<None>

Рисунок 3.5 – Фізична структура таблиці «Специфікація»

Загальну структуру бази даних представлено на рисунку 3.6.

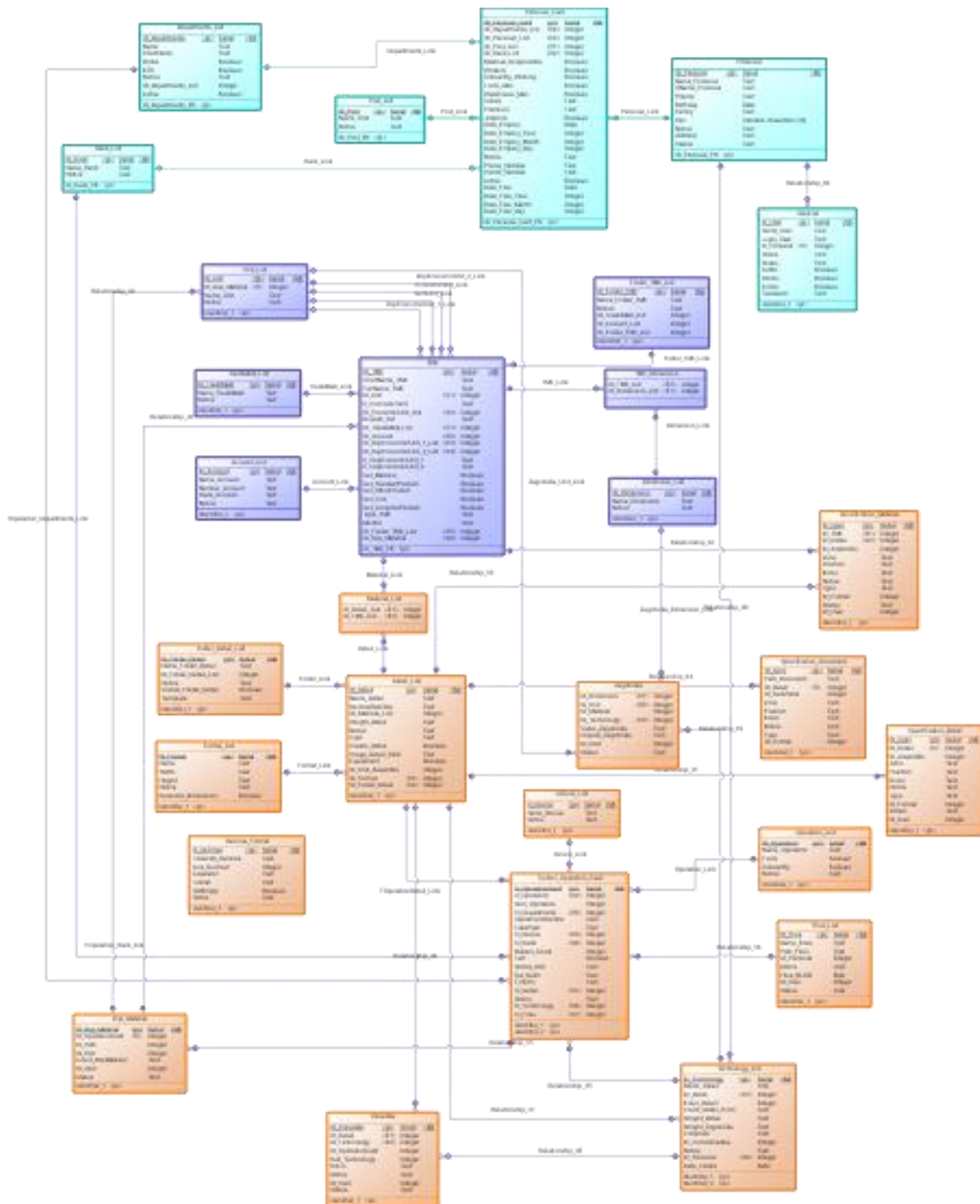


Рисунок 3.6 – Загальна структура бази даних

Як видно з рисунка 3.6, до структури бази даних входить 31 таблиця.

4 РОЗРОБКА ПРОГРАМИ ДЛЯ АВТОМАТИЗАЦІЇ ОБЛІКУ КОНСТРУКТОРСЬКО–ТЕХНОЛОГІЧНОЇ ДОКУМЕНТАЦІЇ

4.1 Вибір мови середовища програмування

C# – це мова програмування, що було створено спеціально для роботи у середовищі Microsoft .NET Framework.

Мову C# було розроблено з урахуванням сильних і слабких особливостей інших мов, зокрема Java та C++.

Ключові особливості мови C#:

- компонентна орієнтованість;
- код зібраний воєдино (декларації і реалізації об'єднані разом);
- уніфікована система типів та їхня безпечність;
- автоматична та мануальна робота за пам'яттю;
- використання єдиної бібліотеки класів – CLR.

Microsoft Visual Studio – лінійка продуктів компанії Microsoft, які містять інтегроване середовище розробки програмного забезпечення, а також низку інших інструментальних засобів. Дані продукти дають можливість розробляти як консольні додатки, так і додатки з графічним інтерфейсом, до того ж, із підтримкою технології Windows Forms, а також веб–сайти, веб–додатки, веб–служби як в рідному, так і в керованому кодах для всіх платформ, які підтримують Microsoft Windows, Windows Mobile, Windows CE, .NETFramework, .NETCompactFramework та MicrosoftSilverlight.

VisualStudio містить редактор вихідного коду з підтримкою технології IntelliSense та можливістю найпростішого рефакторинга коду. Вбудований налагоджувач може працювати як налагоджувач рівня вихідного коду, так і як налагоджувач машинного рівня. Інші вбудовувані інструменти містять редактор форм для спрощення створення графічного інтерфейсу програми, веб–редактор, дизайнер класів і дизайнер схеми бази даних. Visual Studio дає змогу створювати та підключати сторонні додатки (плагіни) з метою

розширення функціональності практично на кожному рівні, включаючи додавання підтримки систем контролю версій вихідного коду (на кшталт Subversion і Visual SourceSafe), додавання нових наборів інструментів (зокрема, для редагування та візуального проєктування коду на предметно-орієнтованих мовах програмування чи інструментів для інших аспектів циклу розробки програмного забезпечення (наприклад, клієнт Team Explorer для роботи з Team Foundation Server).

4.2 Розробка графічної оболонки програми

Для створення інтерфейсу користувача були використані стандартні компоненти, що представлені в інтегрованому середовищі Visual Studio.

Програма має вісім основних форм для роботи з даними, що подані в розділі проєктування структури СКБД.

Форму для відображення даних з довідника ТМЦ зображено на рисунку 4.1.

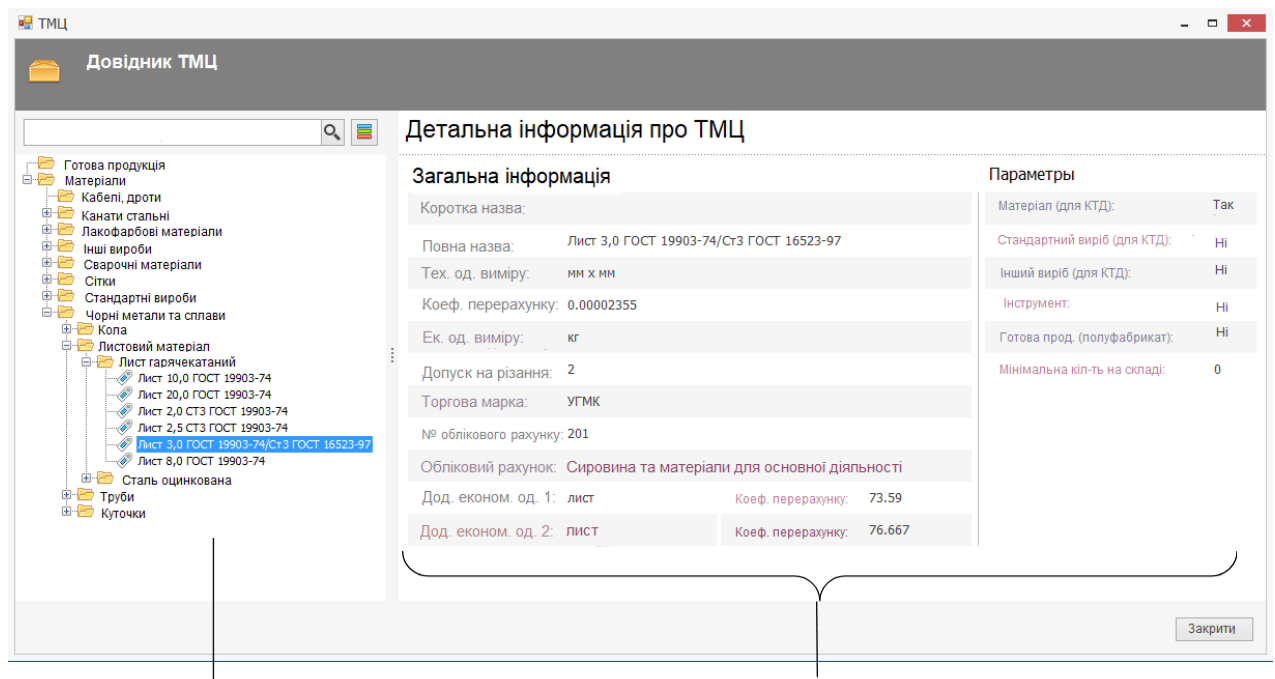


Рисунок 4.1 – Форма для відображення даних з довідника ТМЦ

З рисунку 4.1 визначено, що форма має дві області:

- область зі списком ТМЦ у вигляді дерева;
- область з відображенням характеристик обраної ТМЦ.

В області списку ТМЦ втілено інтерфейс пошуку даних у таблиці ТМЦ за повною чи короткою назвою.

Заповнення списку здійснюється за таким програмним кодом:

```
private void FillTreeView(string filter)
{
    bool field_1 = false;
    bool field_2 = false;
    if (filter == filterCaption) return;
    if (filter.Length == 0) return;
    xtraTabControl1.SelectedTabPage = xtraTabPage1;
    if ((formatDetail.Count > 0) && (filterCaption ==
formatDetail[0].FindText)) field_1 = true;
    else if ((formatDetail.Count > 1) && (filterCaption ==
formatDetail[1].FindText)) field_2 = true;
    foreach (NodesItem item in nodesItem)
    {
        string fText = "";
        bool find = false;
        if (field_1 == true) fText = item.Find_Detail_Field1.ToUpper();
        else if (field_2 == true) fText = item.Find_Detail_Field2.ToUpper();
        if (filter == "") find = true;
        else if (barCheckItem3.Checked == true)
        {
            if (fText.StartsWith(filter.ToUpper())) find = true;
        }
        else if (barCheckItem4.Checked == true)
        {
            if (fText.Contains(filter.ToUpper())) find = true;
        }
    }
}
```

```

    }
    else if (barCheckItem5.Checked == true)
    {
        if (fText.EndsWith(filter.ToUpper())) find = true;
    }
    if (find == true)
    {
        var node = GetNode(item.NodeId);
        treeView1.SelectedNode = node;
        return;
    }
}
}

```

З наведеного коду можна визначити, що список заповнюється за допомогою циклу `foreach`, який бере дані для кожної ітерації з об'єкта `nodesItem`.

Об'єкт `nodesitem` – це список, елементи якого побудовані на основі типу `NodesItem`.

Клас для опису даного типу наведемо нижче:

```

public class NodesItem
{
    public int NodeId { get; set; }
    public int ParentId { get; set; }
    public string NodeName { get; set; }
    public string TypeNode { get; set; }
    public Color nodeColor { get; set; }
    public int imageIndex { get; set; }
    public string Find_Detail_Field1 { get; set; }
    public string Find_Detail_Field2 { get; set; }
    public bool isSelected { get; set; }
}

```

```

public NodesItem(int NodeId, int ParentId, string NodeName, string
TypeNode, Color nodeColor, int imageIndex, string Find_Detail_Field1, string
Find_Detail_Field2, bool isSelected)
{
    this.NodeId = NodeId;
    this.ParentId = ParentId;
    this.NodeName = NodeName;
    this.TypeNode = TypeNode;
    this.nodeColor = nodeColor;
    this.imageIndex = imageIndex;
    this.Find_Detail_Field1 = Find_Detail_Field1;
    this.Find_Detail_Field2 = Find_Detail_Field2;
    this.isSelected = isSelected;
}
public string Name
{
    get { return string.Format("{0}", NodeName); }
}
}

```

Форму для додавання та редагування характеристик ТМЦ наведено на рисунку 4.2.

У даній формі окреслено три області:

- область редагування основних характеристик;
- область для редагування технічної розмірності;
- область для відмітки типу ТМЦ.

У першій області за допомогою текстових полів вводу можна додавати необхідні характеристики ТМЦ.

В області редагування технічної розмірності за допомогою відміток – зазначати, за допомогою яких параметрів вимірюється ТМЦ.

The screenshot shows a software window titled "ТМЦ" (Material Characteristics) with a subtitle "Товарно-матеріальні цінності" (Material Characteristics) and "Інформація о ТМЦ" (Material Characteristics Information). The window is divided into three main sections:

- Загальна інформація (General Information):** Contains fields for "Коротка назва:" (Short name: "Лист 3,0 Ст. 3"), "Повна назва:" (Full name: "Лист 3,0 ГОСТ 19903-74/Ст3 ГОСТ 16523-97"), "Тех. од. виміру:" (Technical unit: "мм"), "Коеф. перерахунку:" (Coefficient: "0.00002355"), "Економ. од. виміру:" (Economic unit: "кг"), "Допуск на порізу:" (Cutting tolerance: "2"), "Торгова марка:" (Trade mark: "УГМК"), "Обліковий рахунок:" (Account: "Сировина та матеріали для основної діяльності"), "Дод. економ. од. 1:" (Additional economic unit 1: "лист"), "Коеф. перерахунку:" (Coefficient: "73.59"), "Дод. економ. од. 2:" (Additional economic unit 2: "лист"), and "Коеф. перерахунку:" (Coefficient: "76.667").
- Технічна розмірність (Technical Dimensions):** A list of checkboxes for "Висота" (Height), "Довжина" (Length), "Кількість" (Quantity), and "Ширина" (Width). "Довжина" and "Ширина" are checked.
- Налаштування (Settings):** A list of checkboxes for "Матеріал (для КТД)" (Material), "Стандартні вироби (для КТД)" (Standard products), "Інші вироби (для КТД)" (Other products), "Інструмент" (Instrument), and "Готова продукція (полуфабрикат)" (Ready-made product). "Матеріал" is checked. Below this is a field for "Мінімальна кіль-ть на складі:" (Minimum quantity in stock) with a value of "0".

At the bottom right, there are "OK" and "Отмена" (Cancel) buttons.

Рисунок 4.2 – Форма для додавання та редагування характеристик ТМЦ

На рисунку 4.2 лист сталі вимірюється параметрами довжини та ширини.

У третій області можна вибрати ознаку ТМЦ. Програма розроблена таким чином, щоб можна було обрати тільки одну ознаку.

Кнопка ОК дозволяє внести введені дані до відповідної таблиці бази даних. Для захисту від помилкового вводу даних, а також від можливих уникань заповнення деяких полів кнопка ОК з'являється тільки після вводу всіх необхідних даних, а не заповнені ще поля підсвічуються червоним.

Для реалізації такої функції було написано такий код:

```
private void textEdit1_TextChanged(object sender, EventArgs e)
{
    TextEdit te = (TextEdit)sender;
    if (te.Text == "")
    {
        label49.ForeColor = Color.Tomato;
        par1 = false;
    }
}
```

```

else
{
    bool nameTest = false;
    String testID = DataBase.Find_Value(String.Format("select
ID_TMC from TMC where ShortName_TMC = '{0}'", te.Text), "ID_TMC");
    if (testID == "0") nameTest = false;
    else if (testID == ID) nameTest = false;
    else nameTest = true;
    if (nameTest == true) {
        label49.ForeColor = Color.Tomato;
        par1 = false; }
    else {
        label49.ForeColor = Color.Gray;
        par1 = true; }
    }
testButtonEnabled();
}

```

Форму для роботи з довідником конструкторсько–технологічної інформації подано на рисунку 4.3.

У даній формі можна бачити три ключові області:

- список КТД у вигляді дерева;
- інформація про деталь;
- специфікація складальної одиниці.

Крім того, на формі присутні чотири закладки:

- конструкторська документація;
- розгорнута специфікація;
- технологія;
- попередній перегляд ескізу деталі.

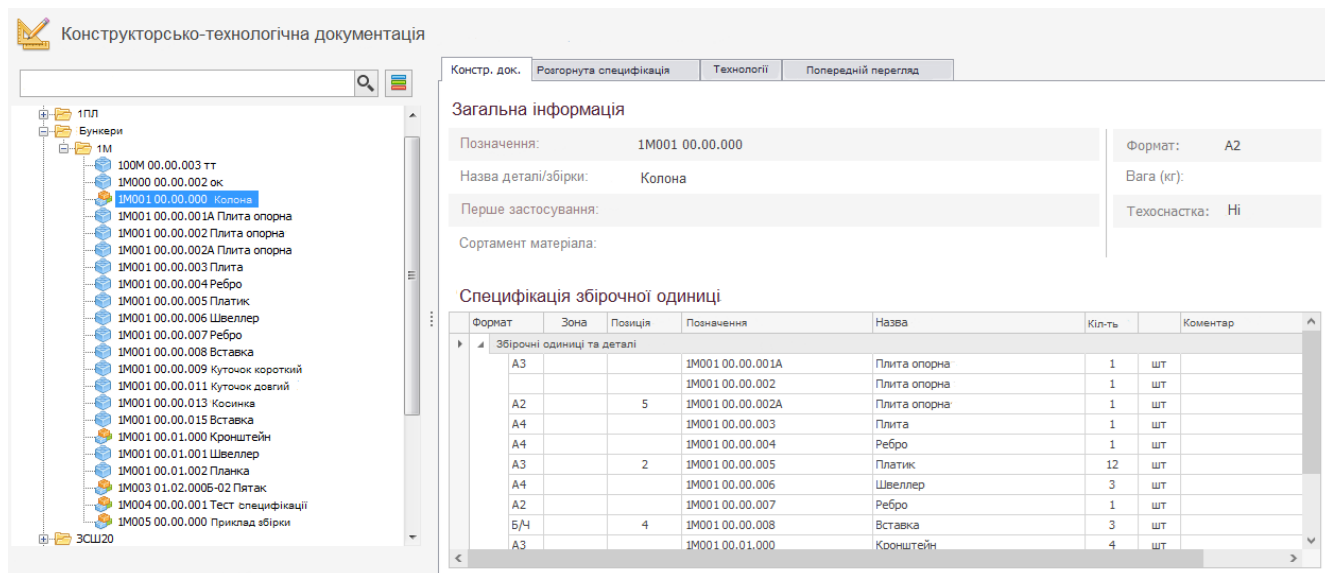


Рисунок 4.3 – Форма для роботи з довідником конструкторсько-технологічної інформації.

На рисунку 4.4 наведено зразок вкладки «Розгорнута специфікація».

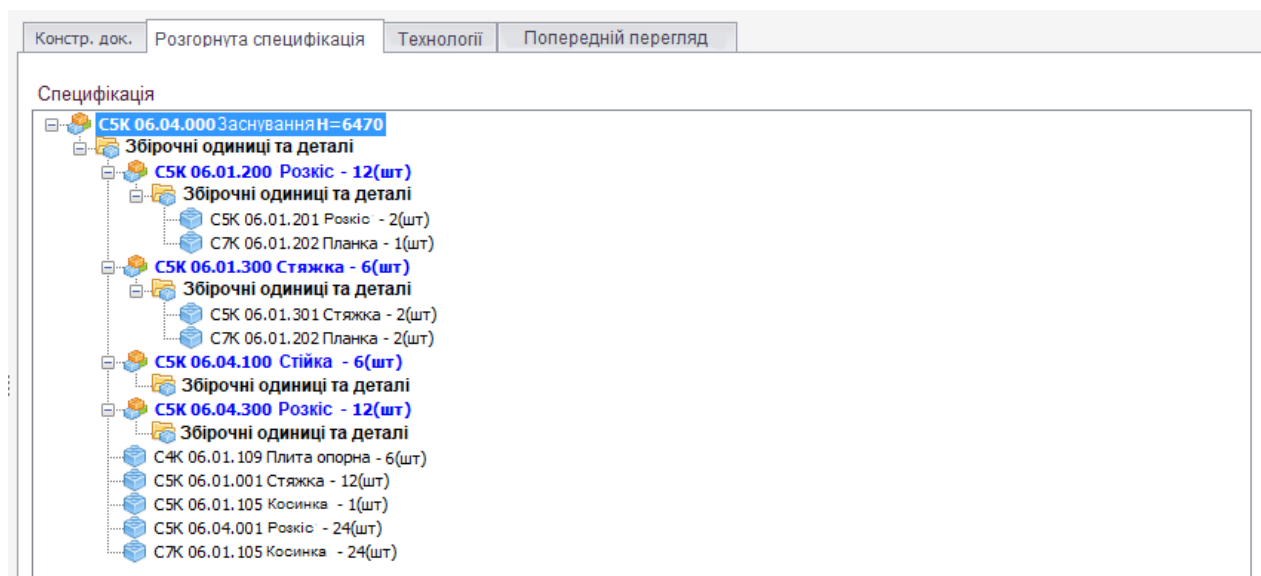


Рисунок 4.4 – Зразок вкладки «Розгорнута специфікація»

Заповнення розгорнутої специфікації реалізується за допомогою коду:

```
private void viewOpenAssemble(FullDetail_ListMembers fullDetail_Info)
```

```
{
    treeView4.BeginUpdate();
```

```

treeView4.Nodes.Clear();
if (fullDetail_Info == null)
{
    treeView4.EndUpdate();
    return;
}
if (fullDetail_Info.Type != "assemble")
{
    treeView4.EndUpdate();
    return;
}
string ID_Assemble = fullDetail_Info.ID_Detail;
OpenAssemble_Tree.Nodes.Clear();
readOpenAssemble(ref OpenAssemble_Tree, ID_Assemble);
Font boldFont = new Font("Tahoma", 8.25f, FontStyle.Bold);
OpenAssemble_Tree.NodeFont = boldFont;
OpenAssemble_Tree.Text = fullDetail_Info.DecimalNumber + " " +
fullDetail_Info.Name_Detail;
OpenAssemble_Tree.ImageIndex = 32;
OpenAssemble_Tree.SelectedImageIndex = 32;
OpenAssemble_Tree.ForeColor = Color.Blue;
treeView4.Nodes.Add(OpenAssemble_Tree);
treeView4.ExpandAll();
treeView4.EndUpdate();
}

```

Вигляд вкладки «Технологія» показано на рисунку 4.5.

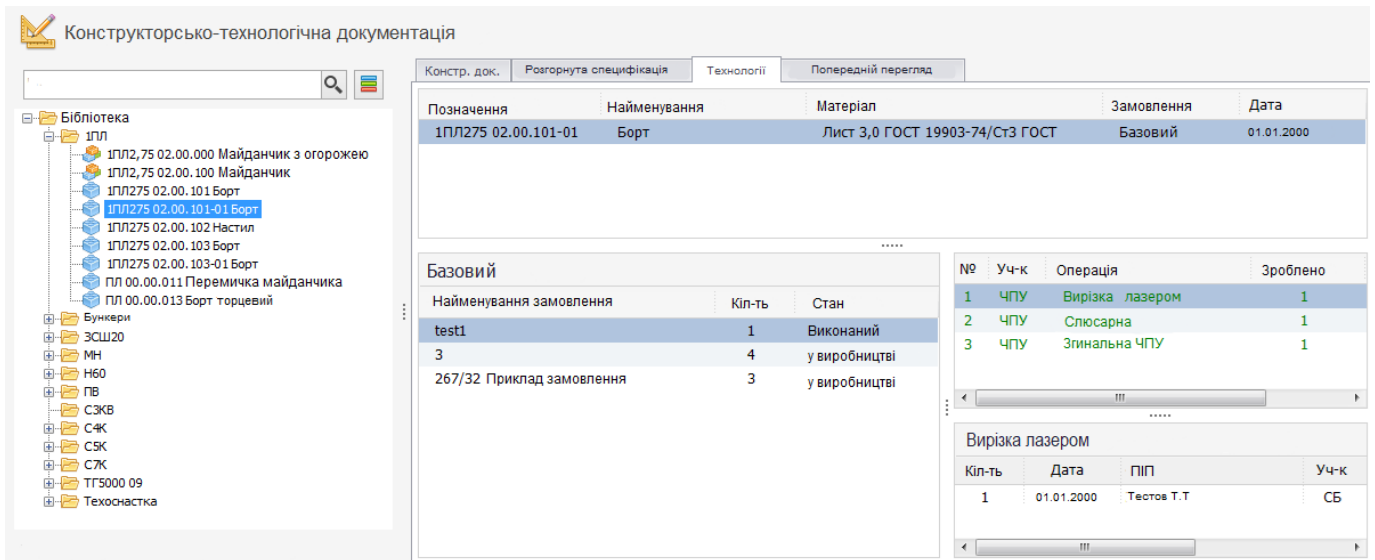


Рисунок 4.5 – Зразок вкладки «Технологія»

На даному рисунку можна побачити, що складовими вкладки «Технологія» є такі області:

- перелік технологій;
- перелік замовлень;
- перелік операцій для обраної технології;
- перелік виконавців для обраної операції.

Розроблення нової технології та редагування вже існуючої реалізується за допомогою форми, зазначеної на рисунку 4.6.

За даним рисунком можна визначити, що в даній формі присутні такі елементи інтерфейсу користувача:

- загальна інформація про замовлення, для якого створюється технологія;
- список технологічних операцій;
- область редагування розміру заготовки та допуску на порізання;
- інформація щодо операції, обраної у відповідному списку;
- інформація про устаткування;
- інформація про додаткові матеріали.

Деталь

Технологія
Технологія виготовлення деталі (збірки)

Загальна інформація

Назва замовлення: Базовий

Назва деталі/збірки: 1ПЛ275 02.00.101 Борт

Основний матеріал: Лист 3,0 ГОСТ 19903-74/Ст3 ГОСТ 16523-97

Кількість деталей: 1 Вага 1 деталі:

Число дет. з 1 загот.: 1 Вага 1 заготовки: 1,10383560

Список технологічних операцій

№	Ділянка	Операц.	Обладнання	Розр	Кіл.роб	Парт	Од.норм	Тпз	Тшт	Р
1	ЧПУ	Вирізка	Лазерний	4	1	Ні	1	0	0.9170	
2	ЧПУ	Слюсарна	Болгарка	3	1	Ні	1	0	0.7336	
3	ЧПУ	Згинальна	Згинальний	4	1	Ні	1	8.5	1.5000	

Ознака готовності: Розробив: Тестов Т.Т

Розмір заготовки та допуск на порізу

Довжина: 252 мм Дод.: 2

Ширина: 186 мм Дод.: 2

Інформація щодо операції

Зачистити після вирізки лазером

Оснащення

Найменування	Кіл-ть

Дод. матеріали

Найменування	Од.в	Кіл-ть
Емаль УРФ-1128 сіра RAL-7000	кг	1

Рисунок 4.6 – Форма для створення нової технології та редагування вже існуючої

Додавання нової операції за допомогою форми зображено на рисунку 4.7.

Картка технологічної операції

Технологічна операція
Детальний опис технологічної операції

Загальна інформація

Номер операції: 1

Ділянка: ЧПУ Ділянка №3

Операція: Вирізка лазером

Обладнання: Лазерний верстат

Опис операції

Вирізати контур деталі та отвору

Нормування

Розряд: 4

Кіл-ть робітників: 1

Тпз: 0

Од.норм.: 1

Довжина лінії різ: 917

Переклад мм в м: 1000

Коефіцієнт складності: 1

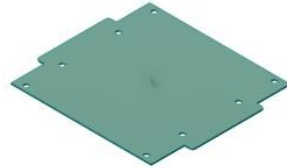
Норма штучного часу: 0,91700

Ознака партії

Додаткова інформація

Матеріали Ескіз

Ескіз: 0С1090.07.00.012 Плита підмоторна.SLDPR



Формула для розрахунку Тшт

$T_{шт} = D_{лр} / 1000 \times K_c$

Рисунок 4.7 – Додавання нової операції

Під час додавання нової операції треба заповнити певні поля:

– загальну інформацію про операцію із зазначенням:

- 1) номера операції;
- 2) найменування ділянки;
- 3) найменування операції;
- 4) найменування обладнання.

– дані щодо нормування, до яких належать:

- 1) розряд;
- 2) кількість робітників;
- 3) підготовчий–заключний час виконання технологічної операції;
- 4) одиниця нормування;
- 5) штучний час виконання технологічної операції (Тшт);
- 6) формула для розрахунку Тшт;
- 7) додаткові дані, що входять до формули.

– список додаткових матеріалів;

– ескіз деталі.

Для створення специфікація було розроблено відповідну форму, її зразок наведено на рисунку 4.8.

Аналізуючи рисунок 4.8, визначаємо, що дана форма дозволяє конструкторові бачити всі позиції специфікації у вигляді списку.

Усі позиції розбиті на групи, кожна з якої розташована у відповідній вкладці. У даному разі таких вкладок шість:

- документація;
- збиральні одиниці та деталі;
- стандартні вироби;
- інші вироби;
- матеріали;
- компоненти.

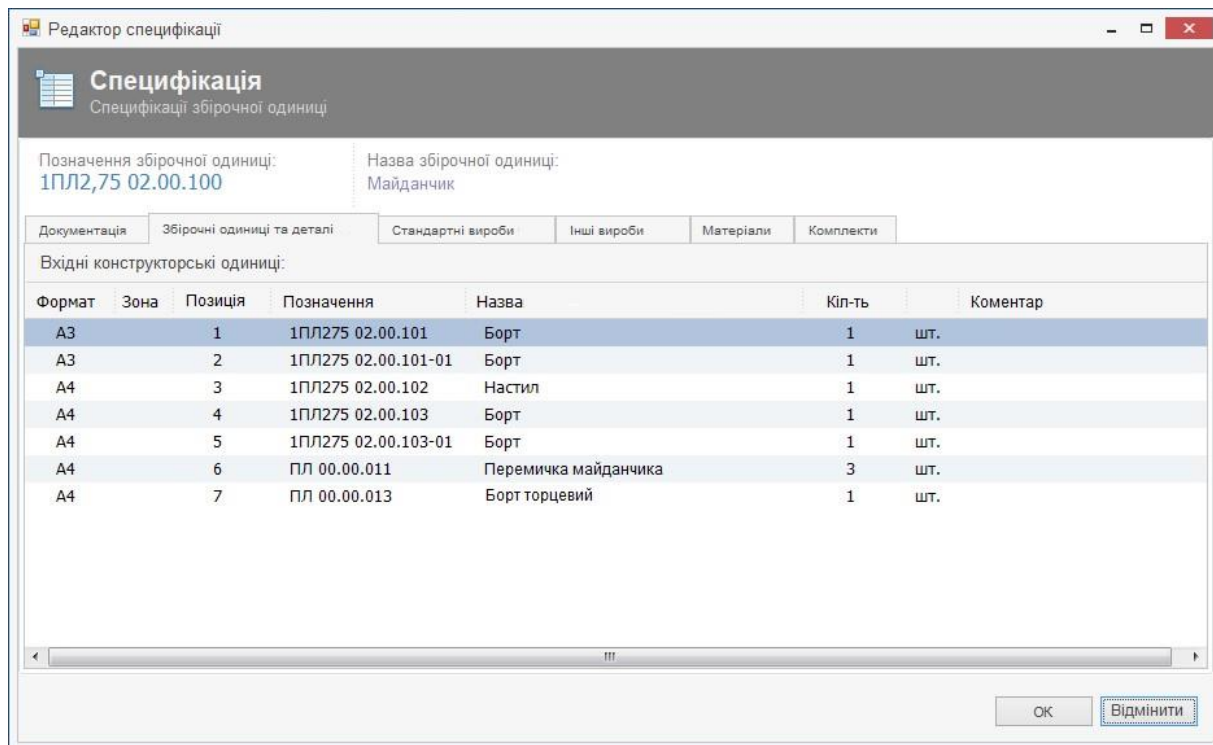


Рисунок 4.8 – Форма для створення специфікації

Під час додавання нової позиції до специфікації використовується спеціальне діалогове вікно, зображене на рисунку 4.9.

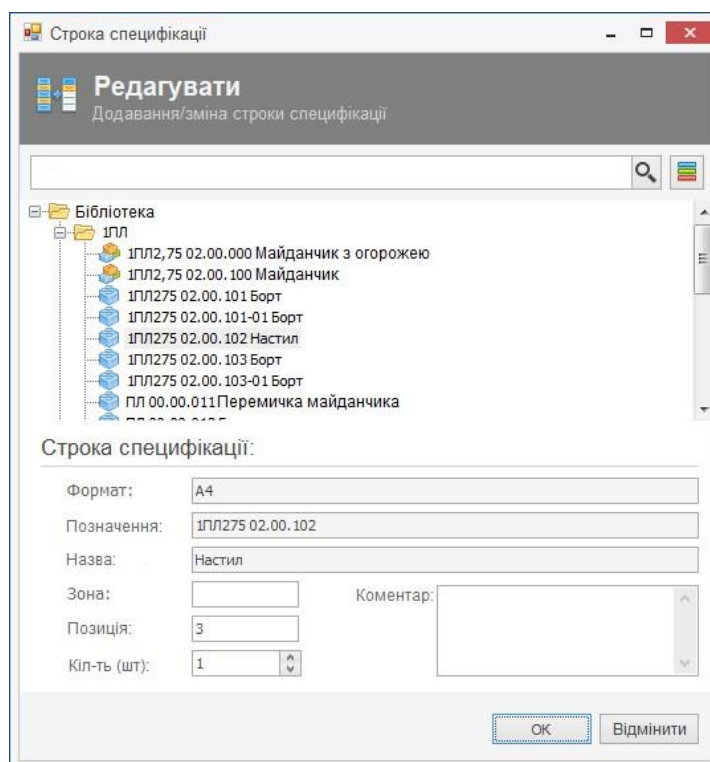


Рисунок 4.9 – Додавання нової позиції до специфікації

У даній формі присутні такі компоненти інтерфейсу, як:

- список КТД;
- поле для відображення формату креслення деталі (не редагується);
- поле для відображення децимального номера (не редагується);
- поле для відображення найменування деталі (не редагується);
- поле для вводу зони;
- поле для вводу позиції;
- поле для вводу кількості деталей;
- поле для вводу коментаря.

4.3 Розробка запитів до бази даних

Для відображення даних із відповідних таблиць у програмі було розроблено кілька запитів до бази даних. Проаналізуємо два основних запити, що дозволять переглянути характеристики деталі та ТМЦ.

З метою отримання інформації про деталь було використано п'ять таблиць:

- Detail_List – довідник деталей;
- TMC – довідник ТМЦ;
- Personal – список співробітників;
- Format_List – довідник форматів;
- Folder_Detail_List – список каталогів.

Побудови запитів здійснюються на базі операторів SELECT та LEFT JOIN. У результаті отримуємо такий запит:

```
string sql = String.Format("select DL.ID_Detail, DL.Name_Detail,
DL.DecimalNumber, DL.ID_Material_List, DL.Count_Detail,
DL.Count_Detail_From, DL.Weight_Detail, DL.Weight_Zagotovka, DL.Notice,
DL.Type, DL.Visible_Detail, DL.Image_Detail_Path, DL.Equipment,
DL.ID_First_Assemble, DL.ID_First_Spec, DL.ID_Format, DL.ID_Folder_Detail,
DL.DecimalN1, DL.DecimalN2, DL.DecimalN3, DL.DecimalN4, DL.DecimalN5,
DL.DecimalN6, DL.DecimalN7, TMC.ShortName_TMC, TMC.FullName_TMC,
```

```
TMC.K_EconomicUnit,          TMC.Dopusk_Cut,          P.Name_Personal,
P.SName_Personal,          P.FName,          F.Name          as          Name_Format,
FDL.Name_Folder_Detail, FDL.Template, FA.Name_detail as Name_Assemble
from Detail_List as DL LEFT JOIN TMC as TMC on DL.ID_Material_List =
TMC.ID_TMC LEFT JOIN Personal as P on DL.ID_Personal = P.ID_Personal
LEFT JOIN Format_List as F on DL.ID_Format = F.ID_Format LEFT JOIN
Folder_Detail_List as FDL on DL.ID_Folder_Detail = FDL.ID_Folder_Detail
LEFT JOIN Detail_List as FA on DL.ID_First_Assemble = FA.ID_Detail WHERE
DL.ID_Detail = {0};" , ID_Detail);
```

Для отримання інформації про деталь виконано п'ять таблиць:

- Detail_List – довідник деталей;
- TMC – довідник ТМЦ;
- Unit_List – довідник економічних одиниць;
- TradeMark_List – довідник торговельних марок;
- Account_List – довідник рахунків.

Відображення характеристик ТМЦ здійснюється на основі такого запиту:

```
string sql = String.Format("select TMC.ID_TMC, TMC.ShortName_TMC,
TMC.FullName_TMC,          TMC.ID_Unit,          TMC.K_EconomicUnit,
TMC.ID_EconomicUnit_List,TMC.Dopusk_Cut,          TMC.ID_TradeMark_List,
TMC.ID_Account,TMC.ID_DopEconomicUnit_1_List,TMC.ID_DopEconomicUnit_
t_2_List,          TMC.K_DopEconomicUnit_1,          "TMC.K_DopEconomicUnit_2,
TMC.Set_Material,          TMC.Set_StandartProduct,          TMC.Set_OtherProduct,
TMC.Set_Tool,          TMC.Set_CompliteProduct,          Type_TMC,          TMC.MinKol,
TMC.ID_Folder_TMC_List,          UL.Name_Unit,          ecoUL.Name_Unit          as
Name_EconomicUnit, DopEcoUL_1.Name_Unit as Name_DopEconomicUnit_1,
DopEcoUL_2.Name_Unit as Name_DopEconomicUnit_2, TM.Name_TradeMark,
AL.Name_Account, AL.Number_Account, AL.State_Account from TMC LEFT
JOIN Unit_List as UL on TMC.ID_Unit = UL.ID_Unit LEFT JOIN Unit_List as
ecoUL on TMC.ID_EconomicUnit_List = ecoUL.ID_Unit LEFT JOIN Unit_List as
DopEcoUL_1 on TMC.ID_DopEconomicUnit_1_List = DopEcoUL_1.ID_Unit
```

```
LEFT JOIN Unit_List as DopEcoUL_2 on TMC.ID_DopEconomicUnit_2_List =
DopEcoUL_2.ID_Unit LEFT JOIN TradeMark_List as TM on
TMC.ID_TradeMark_List = TM.ID_TradeMark LEFT JOIN Account_List as AL
on TMC.ID_Account = AL.ID_Account WHERE TMC.ID_TMC = {0};",
ID_TMC);
```

4.4 Оцінка складності програмного засобу

Як правило, одній формі програми відповідає кілька транзакцій. Типовий екран містить кілька зовнішніх запитів, які супроводжують зовнішнє введення даних.

Введення, виведення та запити належать до категорії "транзакція". Транзакцією називається елементарний процес, який переміщує дані між зовнішнім середовищем і програмним додатком. У своїй роботі транзакції використовують як внутрішні, так і зовнішні файли.

Зовнішнє введення є елементарним процесом, який переміщає дані з зовнішнього середовища до додатка. Дані можуть надходити з екрана введення або з іншої програми, крім того, можуть використовуватися для оновлення внутрішніх логічних файлів. До того ж, дані можуть містити як ділову інформацію, так і керівну, проте, керівні дані не повинні модифікувати внутрішній логічний файл.

Зовнішнє виведення є елементарним процесом, який переміщає дані, що були обчислені в додатку, до зовнішнього середовища. До того ж, у цьому процесі можуть оновлюватися внутрішні логічні файли. Дані створюють звіти або вихідні файли, що надсилаються до інших додатків. Звіти та файли створюються на основі як внутрішніх логічних файлів, так і зовнішніх інтерфейсних файлів. Додатково цей процес може застосовувати дані, що вводяться, їх утворюють критерії пошуку та параметри, що не підтримуються внутрішніми логічними файлами. Дані для введення надходять ззовні, проте носять тимчасовий характер, тому не зберігаються у внутрішньому логічному файлі.

Зовнішній запит є елементарним процесом, який працює з вхідними та вихідними даними. Його результатом є дані, що повертаються з внутрішніх логічних файлів і зовнішніх інтерфейсних. Вхідна частина процесу не модифікує внутрішні логічні файли, а вихідна – не несе даних, які обчислюються додатком (у цьому і полягає відмінність запиту від звичайного виведення).

Внутрішній логічний файл є групою логічно пов'язаних даних, яка розміщена всередині додатка й обслуговується через зовнішні введення.

Зовнішній інтерфейсний файл є групою логічно пов'язаних даних, яка розміщена всередині іншого додатка і підтримується ним.

Кожній із визначених характеристик у відповідність ставиться складність. Для цього характеристиці потрібно призначити низький, середній або високий ранг, а потім сформуванати числову оцінку рангу.

У програмі з графічним інтерфейсом генеруються 3 типи повідомлень: власне повідомлення, повідомлення про помилку, повідомлення підтвердження. Повідомлення про помилку (на зразок пароля), а також повідомлення підтвердження (на кшталт "Ви дійсно хочете видалити клієнта?") вказують, що сталася помилка або процес можна завершити. Такого роду повідомлення не утворюють самостійного процесу, тому що є частиною іншого процесу, тобто вважаються елементом даних транзакції.

З іншого боку, повідомлення – це незалежні елементарні процеси. Зокрема, під час спроби отримати з банкомата суму грошей, яка перевищує їхню кількість на рахунку, генерується повідомлення "Не вистачає коштів для завершення транзакції". Воно є результатом зчитування інформації з файлу рахунка та формуванням висновку. Повідомлення розглядається як зовнішнє виведення.

У таблиці 4.1 наведено дані для визначення складності зовнішніх уведень.

Таблиця 4.1 – Ранг і оцінка складності зовнішніх уведень

Посилання	Елементи даних		
	1–4	5–15	>15
0–1	Низький (3)	Низький (3)	Середній (4)
2	Низький (3)	Середній (4)	Високий (6)
>2	Середній (4)	Високий (6)	Високий (6)

У таблиці 4.2. наведено дані для визначення складності зовнішніх виведень

Таблиця 4.2 – Ранг і оцінка складності зовнішніх виведень

Посилання	Елементи даних		
	1–4	5–19	>19
0–1	Низький (4)	Низький (4)	Середній (5)
2–3	Низький (4)	Середній (5)	Високий (7)
>3	Середній (5)	Високий (7)	Високий (7)

У таблиці 4.3 наведено дані для визначення складності зовнішніх запитів.

Таблиця 4.3 – Ранг і оцінка складності зовнішніх запитів

Посилання	Елементи даних		
	1–4	5–19	>19
0–1	Низький (3)	Низький (3)	Середній (4)
2–3	Низький (3)	Середній (4)	Високий (6)
>3	Середній (4)	Високий (6)	Високий (6)

Зауважимо, якщо в зовнішньому запиті посилання на файл використовується як на етапі введення, так і на етапі виведення, воно враховується тільки один раз. Аналогічне правило поширюється і на елемент даних (одноразовий облік).

У таблиці 4.4 наведено дані для визначення складності інтерфейсних файлів.

Таблиця 4.4 – Ранг і оцінка складності інтерфейсних файлів

Посилання	Елементи даних		
	1–19	20–50	>50
0–1	Низький (5)	Низький (5)	Середній (7)
2–5	Низький (5)	Середній (7)	Високий (10)
>5	Середній (7)	Високий (10)	Високий (10)

У таблиці 4.5. наведено дані для визначення складності внутрішніх логічних файлів.

Таблиця 4.5 – Ранг і оцінка складності внутрішніх логічних файлів

Посилання	Елементи даних		
	1–19	20–50	>50
0–1	Низький (7)	Низький (7)	Середній (10)
2–5	Низький (7)	Середній (10)	Високий (15)
>5	Середній (10)	Високий (15)	Високий (15)

Кількісне значення характеристик кожного виду визначається шляхом підрахунку відповідних елементів графічного інтерфейсу основних вікон програми.

Більш детально розглянемо форму для відображення даних із довідника ТМЦ, що зображено на рисунку 4.1. На даній формі можна побачити тільки зовнішні виведення: один список ТМЦ та 19 полів для відображення інформації з БД.

Із SQL запиту визначаємо, що в ньому використовується 19 елементів даних:

```
sql = String.Format("INSERT INTO TMC (ShortName_TMC,
FullName_TMC, ID_Unit, K_EconomicUnit, ID_EconomicUnit_List, Dopusk_Cut,
ID_TradeMark_List, ID_Account, ID_DopEconomicUnit_1_List,
ID_DopEconomicUnit_2_List, K_DopEconomicUnit_1, K_DopEconomicUnit_2,
Set_Material, Set_StandartProduct, Set_OtherProduct, Set_Tool,
Set_CompliteProduct, Type_TMC, MinKol, ID_Folder_TMC_List ) VALUES
('{0}', '{1}', '{2}', '{3}', '{4}', '{5}', '{6}', '{7}', '{8}', '{9}', '{10}', '{11}', '{12}',
'{13}', '{14}', '{15}', '{16}', '{17}', '{18}', '{19}' ) RETURNING ID_TMC;",
```

item.ShortName_TMC, item.FullName_TMC, item.ID_Unit,
 item.K_EconomicUnit, item.ID_EconomicUnit_List, item.Dopusk_Cut,
 item.ID_TradeMark_List, item.ID_Account, item.ID_DopEconomicUnit_1_List,
 item.ID_DopEconomicUnit_2_List, item.K_DopEconomicUnit_1,
 item.K_DopEconomicUnit_2, item.Set_Material, item.Set_StandartProduct,
 item.Set_OtherProduct, item.Set_Tool, item.Set_CompliteProduct,
 item.Type_TMC, item.MinKol, item.ID_Folder_TMC_List);

Для отримання даних використовуються один зовнішній запит, який отримує інформацію від восьми таблиць. Отже, отримані дані занесемо до таблиці 4.6.

Таблиця 4.6 – Оцінка складності форми відображення даних із довідника ТМЦ

Вид характеристики	Кількість	Ранг	Оцінка
Зовнішні вводи	0		0
Зовнішні виводи	20	Середній (5)	100
Зовнішні запити	19	Середній (4)	76
Логічні файли	8	Низький (7)	56
Зовнішні інтерфейсні файли	0		0
Всього			232

Опрацюємо форму для додавання та редагування характеристик ТМЦ, яку зображено на рисунку 4.2. На ній можна бачити такі зовнішні введення: 12 полів загальної інформації, 4 поля для редагування технічної розмірності, 6 полів додаткової інформації. Крім того, можна бачити випадні списки, з яких користувач може обирати дані для введення:

- коефіцієнт перетворення технічної одиниці вимірювання на економічну;
- список з економічними одиницями вимірювання;
- список торговельних марок;
- список рахунків;
- список із додатковими економічними одиницями вимірювання 1;

- список із додатковими економічними одиницями вимірювання 2;
- список коефіцієнтів перетворення технічної одиниці вимірювання на додаткову економічну 1;
- список коефіцієнтів перетворення технічної одиниці вимірювання на додаткову економічну 2;
- мінімальна кількість ТМЦ на складі.

Отже, загальна кількість зовнішніх виведень – 9, а кількість таблиць, з яких отримуються дані – 8.

У запитах використано 29 елементів даних.

Для зберігання даних застосовують один зовнішній запит, який додає інформацію до однієї таблиці. До таблиці 4.7 занесемо отримані дані.

Таблиця 4.7 – Оцінка складності форми для додавання та редагування характеристик ТМЦ

Вид характеристики	Кількість	Ранг	Оцінка
Зовнішні вводи	22	Середній (4)	88
Зовнішні виводи	9	Низький (4)	36
Зовнішні запити	29	Середній (4)	116
Логічні файли	8	Низький (7)	56
Зовнішні інтерфейсні файли	0		0
Всього			296

Після збору всієї необхідної інформації розпочнемо розрахунки метрики – кількості функціональних показників *FP* (Function Points).

Обчислимо кількість функціональних показників за формулою (4.1)

$$FP = \text{Загальна кількість} \cdot (0,65 + 0,01 \times \sum F_i), \quad (4.1)$$

де F_i – коефіцієнти регулювання складності.

Кожен коефіцієнт може приймати певні значення: 0 – немає впливу, 1 – випадкове, 2 – невелике, 3 – середнє, 4 – важливе, 5 – основне.

Значення було вибрано емпірично на основі відповідей на 14 питань, які характеризують системні параметри програми (таблиця 4.8).

Таблиця 4.8 – Визначення системних параметрів програми

№	Системний параметр	Опис	Значення
1	Передача даних	Скільки засобів зв'язку потрібно для передачі або обміну інформацією з додатком або системою?	3
2	Розподілена обробка даних	Як обробляються розподілені дані і функції обробки?	3
3	Продуктивність	Чи потребує користувач у фіксації часу відповіді або продуктивності?	5
4	Поширеність використовуваної конфігурації	Наскільки поширена поточна апаратна платформа, на якій буде виконуватися додаток?	4
5	Швидкість транзакцій	Як часто виконуються транзакції? (щодня, щотижня, щомісяця)	5
6	Оперативне введення даних	Який відсоток інформації треба вводити в режимі онлайн?	4
7	Ефективність роботи кінцевого користувача	Додаток проектувався для забезпечення ефективної роботи кінцевого користувача?	4
8	Оперативне оновлення	Як багато внутрішніх файлів оновлюється в онлайнній транзакції?	4
9	Складність обробки	Чи виконує додаток інтенсивну логічну або математичну обробку?	3
10	Повторна використовуваність	Додаток розроблявся для задоволення вимог одного чи багатьох користувачів?	4
11	Легкість інсталяції	Наскільки важкі перетворення та інсталяція програми?	5
12	Легкість експлуатації	Наскільки ефективні та / або автоматизовані процедури запуску, резервування та відновлення?	5
13	Різноманітні умови розміщення	Чи була спроектована, розроблена і підтримана можливість інсталяції програми в різних місцях для різних організацій?	5
14	Простота змін	Чи була спроектована, розроблена і підтримана в додатку простота змін?	4

Отже, виконаємо розрахунок функціональних метрик:

$$FP = 528 \cdot (0,65 + 0,01 \times (3 + 3 + 5 + 4 + 5 + 4 + 4 + 4 + 3 + 4 + 5 + 5 + 5 + 4)) = 649,44.$$

Отримане значення підтверджує значно високу складність програми, що було розроблено в рамках даної роботи.

4.5 Забезпечення безпечних умов праці у лабораторій при проведенні досліджень

Розміри лабораторії, в якій виконувалася робота, складають 5 м × 6 м. Робоче місце складається зі стола, стільця і персонального комп'ютера. У приміщенні працює 4 людини. Площа приміщення 30 м², об'єм – 90 м³. Згідно ДСанПіН 3.3.2.007–98 площа на одне робоче місце має становити не менше 6 м², а об'єм – 20 м³. Для даного приміщення робоча площа і об'єм на одну людину відповідає нормам, так як в нашому випадку площа на одне робоче місце становить 10 м², а об'єм – 30 м³.

Живлення комп'ютерів здійснюється від трифазної чотирипровідної електричної мережі змінного струму з глухо–заземленою нейтраллю і напругою 220 В, частотою 50 Гц.

Згідно НПАОП 40.1–1.21–98 лабораторію можна віднести до категорії без підвищеної небезпеки, так як в приміщенні відсутні чинники, які викликають підвищену або особливу небезпеку.

Для створення безпечних умов праці необхідно провести ряд організаційних і технічних заходів. Згідно НПАОП 40.1–1.32–01 для запобігання ураження людини електричним струмом в приміщенні застосовується система занулення.

Згідно вимог НПАОП 0.00–4.12–05 необхідно провести увідний, первинний на робочому місці, повторний, цільовий та позаплановий інструктажі. Зміст інструктажу відповідає вимогам НПАОП 0.00–4.12–05.

Інструктаж відзначається в відповідних журналах з підписами інструктованих і інструктора.

Робота в лабораторії проводиться сидячи і не вимагає фізичного напруження. Тому вона відноситься до категорії Іа (легкі фізичні роботи, енергозатрати до 120 ккал/ч). З метою забезпечити комфортні умови для працівників та згідно з ДСН 3.3.6.042–99 в приміщенні встановлені наступні метеорологічні параметри:

– для холодного періоду:

- 1) температура повітря від 22 °С до 24 °С;
- 2) вологість повітря від 40 % до 60 %;
- 3) швидкість руху повітря оптимальна до 0,1 м/с;

– для теплого періоду року:

- 1) температура повітря від 23 °С до 25 °С;
- 2) вологість повітря від 40 % до 60 %;
- 3) швидкість руху повітря оптимальна до 0,1 м/с.

Для освітлення робочих місць і приміщення в цілому застосовується як природне бічне освітлення, так і штучне освітлення.

Приміщення з ЕОМ повинні мати природне і штучне освітлення відповідно до ДБН В.25–28–2006 «Природне і штучне освітлення». Природне світло повинно проникати через бічні світлові прорізи, зорієнтовані, як правило, на північ або північний схід, і забезпечувати коефіцієнт природної освітленості (КПО) не нижче 1,5 %:

$$e^{IV} = 1.35,$$

де $e^{IV}_{\text{норм}}$ – нормоване значення КПО для 4–го поясу світлового клімату.

Згідно ДСН 3.3.6.037–99 рівень шуму в лабораторії не перевищує 50 дБ.

Загальний рівень штучного освітлення приміщення можна перевірити за допомогою методу питомої потужності.

Розрахункова формула методу [13]:

$$W = \frac{W_{\Sigma}}{S}, \quad (4.1)$$

де W – питома потужність, Вт/м²;

S – площа приміщення, м²;

W_{Σ} – загальна потужність освітлювальної установки, Вт.

Загальна потужність освітлювальної установки розраховується за формулою

$$W_{\Sigma} = W_{ce} \cdot n_{ce}, \quad (4.2)$$

де W_{ce} – потужність одного світильника, $W_{ce} = 80$ Вт;

n_{ce} – кількість світильників в приміщенні, $n_{ce} = 4$ шт.

Розглянуте приміщення має площу 30 м², в якому розташовано шість світильників потужністю 80 Вт

$$W_{\Sigma} = 4 \cdot 80 = 320 \text{ Вт},$$

$$W = \frac{320}{30} = 11 \text{ Вт/м}^2.$$

Табличне значення для отриманого результат освітленість складе 200 лк, коли згідно стандарту ДБН В.2.5–28–2006. в лабораторії освітленість повинна бути 300–500 лк. Для отримання освітленості в 400 лк необхідна питома потужність 21 Вт/м².

Для поліпшення умов роботи в лабораторії необхідно в денний час застосовувати додаткове освітлення.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи було проаналізовано технічне завдання, визначені основні вимоги програми, що розробляється:

- створення картки конструкторської одиниці;
- створення специфікації конструкторських одиниць;
- створення технології виготовлення деталі;
- формування технологічних маршрутних карт.

Проведено аналіз загальних принципів організації технологічного процесу виготовлення виробу.

Визначено основні вимоги до єдиного інформаційного простору технологічної підготовки виробництва.

Проведено аналіз аналогічних програмних модулів. Серед використовуваних у світі PDM–систем, що відповідають сучасним вимогам, одним із найближчих аналогів програми, що розробляється, обрано PDM SmarTeam.

Розроблено архітектуру програмного модуля та виконано моделювання його роботи за допомогою мови UML. Виконавши аналіз вихідних даних, та моделювання роботи програмного модуля, встановили основний набір даних, що потребує зберігання та обробці:

- товарно–матеріальна цінність;
- деталь;
- картка технологічної операції;
- технологія виготовлення деталі;
- специфікація.

Моделювання структури даних було виконано в автоматизованому середовищі PowerDesigner.

Програма написана на мові програмування C# з використанням Microsoft Visual Studio. Програма має вісім основних форм для роботи з даними, що представлені в розділі проектування структури СКБД.

Для відображення даних з відповідних таблиць в програмі були розроблені SQL–запити до бази даних.

В результаті проведеної роботи можна зробити висновок, що єдиний інформаційний простір створює основу для успішного комплексного функціонування АТПВКІВ, для ефективної організації спільної, узгодженої роботи конструкторів, технологів та інших фахівців ТПВ підприємства. В результаті забезпечуються:

- прискорення процесів ТПВ за рахунок паралельного виконання робіт та електронного обміну даними між фахівцями;
- підвищення якості та достовірності інформації за рахунок прозорості системи і взаємоконтролю учасників процесів проектування;
- накопичення і збереження інформації в електронному вигляді;
- відсутність непотрібного дублювання інформації;
- гнучкість створюваної АТПВКІВ, зручність її розвитку та адаптації до мінливих умов виробництва.

Проведено забезпечення безпечних умов праці в лабораторії, де проводились дослідження. В результаті чого були запропоновані шляхи покращення умов праці та подолання впливу шкідливих чинників на працеспроможність та результативність роботи працівників в лабораторії, що безпосередньо впливає на кінцевий результат роботи та забезпечує безпеку життєдіяльності.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки з підготовки й оформлення кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології, освітньо-професійних програм: «Автоматизоване управління технологічними процесами», «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» / Упоряд. І.Ш. Невлюдов, Р.В. Артюх, Н.П. Демська, В.В. Євсєєв, О.І. Филипенко, О.М. Цимбал. – Харків: ХНУРЕ, 2021. – 50 с.

2. Дипломне проектування для студентів усіх форм навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» : навч. посіб. / за ред. І.Ш. Невлюдов, А.О. Андрусевич, О.В. Токарева, Г.В. Пономарьова. Київ, 2016. 320 с.

3. ДСТУ 3008:2015 Інформація та документація «Звіти у сфері науки і техніки». Структура та правила оформлювання. / В. Земцева; Ю. Поліщук, канд. фіз.-мат. наук; Р. Санченко, канд. техн. наук; Л. Шрамко; А. Ямчук (науковий керівник) ДП «УкрНДНЦ» від 22 червня 2015р. № 61 з 2017-07-01.

4. Положення про кваліфікаційну роботу здобувача вищої освіти на другому (магістерському) рівні [Електронний ресурс] : Наказ ХНУРЕ від 06 травня 2021 р. No 143. – Режим доступу: https://nure.ua/wpcontent/uploads/Main_Docs_NURE/143-vid-06.05.2021-pro-vvedennja-v-dijurishennja-vchenoi-radi-universitetu.pdf.

5. Rodion Shmyha, Rauf Allakhveranov / Unified Information Space for Technological Preparation of Production / X International Scientific and Practical Conference «MODERN PROBLEMS OF SCIENCE, EDUCATION AND SOCIETY» – Kyiv, Ukraine.: 2023. PP. 380–385.

6. Невлюдов І.Ш. Основи виробництва електронних апаратів / І.Ш. Невлюдов. – Харків: Компанія СМІТ, 2005. – 592 с.

6. Невлюдов І.Ш. Технічні засоби автоматизації: Підручник / І.Ш. Невлюдов, А.О. Андрусевич, О.І. Филипенко, Н.П. Демська, С.П. Новоселов. – Кривий Ріг : Криворізький коледж НАУ, 2019. – 366 с.

7. Невлюдов І.Ш. Людино–машинний інтерфейс в технічних засобах автоматизації: Навчальний посібник / І.Ш. Невлюдов, О.І. Филипенко, Б.О. Шостак. – Харків : «ХТМТ», 2019. – 244 с.

8. Костюк В. І. Робототехніка і мехатроніка / В. І. Костюк, Г. О. Спиноу. – К.: Вища школа, 1994. – 446 с.

9. System for automatic planning, execution and evaluation of assembly sequences for industrial robots / U. Thomas, F.M. Wahl. – Institute of Robotics and Process Control, Technical University of Braunschweig, Germany, 2007. – 23 с.

10. Карнаух, С. Г. Розрахунки механічних передач / С. Г. Карнаух, Н. В. Чоста, – К.: ДДМА, 2008. – 100 с.

11. Антоненко, І. І. Технічна механіка / І. І. Антоненко, С. М. Перга – К.: КДПУ, 2017. – 191 с.

12. Діагностика та контроль робочих процесів: навч. посібник для студентів спеціальності «Прикладна механіка» денної та дистанційної форм навчання / В. М. Доля – Харків: НТУ «ХПІ», 2019. – 129 с.

13. Комплекс навчально–методичного забезпечення навчальної дисципліни "Організація керування умовами праці" підготовки освітнього рівня бакалавр усіх спеціальностей та усіх напрямів університету [Електронний ресурс] / ХНУРЕ; розроб.: Т. Є. Стиценко, Г. В. Пронюк, Н. М. Сердюк. – Харків, 2017. – 108 с.