

Центральноукраїнський національний технічний університет
Факультет будівництва, транспорту та енергетики
Кафедра «Автоматизації виробничих процесів»

«Допущено до захисту»
Зав. кафедрою АВП
к.т.н., доцент
Олександр Дідик
«___» _____ 20__ р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему:
«Інформаційна система для автоматизованого виготовлення меблів»

Виконав здобувач вищої освіти
4 курсу, групи АК-19
ОПП «Автоматизація та
комп'ютерно-інтегровані технології»
спеціальності 151 «Автоматизація та
комп'ютерно-інтегровані технології»
_____ Мороз І.А.
«___» _____ 20__ р.

Керівник роботи
професор, к.т.н.
_____ Віктор Каліч
«___» _____ 20__ р.

Рецензент
доцент, к.т.н. кафедри КБПЗ
_____ Юрій Пархоменко
«___» _____ 20__ р.

м. Кропивницький

Центральноукраїнський національний технічний університет

Факультет будівництва, транспорту та енергетики

Кафедра автоматизації виробничих процесів

Рівень вищої освіти перший (бакалаврський) рівень

Галузь знань 15 «Автоматизація та приладобудування»

Спеціальність 151 «Автоматизація та комп'ютерно-інтегровані технології»

Освітньо-професійна програма «Автоматизація та комп'ютерно-інтегровані технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри Дідик О.К.

“___” _____ 20__ року

**ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТА
ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ
ЗДОБУВАЧА ВИЩОЇ ОСВІТИ**

Мороза Іллі Андрійовича

(прізвище, ім'я, по батькові)

1. Тема роботи **«Інформаційна система для автоматизованого виготовлення меблів»**

2. Керівник роботи

Каліч Віктор Михайлович, к.т.н., професор

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

3. Строк подання роботи до захисту 23.06.2023 року

4. Мета та завдання випускної кваліфікаційної роботи

Зниження економічних витрат на матеріали для виготовлення меблів за рахунок оптимізації їх використання. Для досягнення мети необхідно розв'язати наступні задачі: обґрунтувати необхідність розробки інформаційної системи; розробити інформаційне забезпечення системи; виконати опис фізичної реалізації бази даних.

5. Перелік графічного матеріалу організаційна структура підприємства; контекстні діаграми; діаграми потоків даних; етапи розробки АІС; структура АІС; концептуальна схема структури АІС; фізична модель інформаційної бази.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	<i>Опис підприємства. Функції та організаційна структура. Опис потоків даних.</i>	11.04.2023	
2	<i>Огляд існуючих технічних рішень та виявлення їх переваг і недоліків. Обґрунтування необхідності розробки інформаційної системи.</i>	27.04.2023	
3	<i>Вимоги до системи. Етапи розробки автоматизованої інформаційної системи.</i>	11.05.2023	
4	<i>Розробка архітектури побудови та платформи реалізації АІС. Інформаційне забезпечення.</i>	29.05.2023	
5	<i>Опис фізичної реалізації бази даних. Опис структури програмного забезпечення. Алгоритмізація типових інформаційних запитів.</i>	12.06.2023	
6	<i>Оформлення пояснювальної записки бакалаврської роботи та графічної частини</i>	20.06.2023	

Дата видачі завдання

«___» _____ 20__р.

Підпис керівника

(прізвище та ініціали)

Завдання прийнято до виконання

«___» _____ 20__р.

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

В бакалаврській роботі представлено проект інформаційної системи «Автоматизована інформаційна система підприємства з виготовлення корпусних меблів» (АІС «Корпусні меблі»).

Виконано аналіз предметної області; розроблено функціональну схему АІС «Корпусні меблі»; розроблено концепцію, обрано архітектуру побудови і платформу реалізації системи; спроектовано концептуальну модель АІС «Корпусні меблі»; спроектовано логічну модель системи АІС «Корпусні меблі» на основі концептуальної моделі; визначено фізичну структуру серверу баз даних.

Результатом роботи стала АІС «Корпусні меблі», що забезпечує можливість створення замовлення і плану розкрою листа матеріалу, ведення документів про виконання робіт і розрахунку з замовником, пошуку даних за заданими користувачем критеріями, а також забезпечує можливість збереження і друку створюваних документів, включаючи план розкрою.

ABSTRACT

The bachelor's thesis presents the project of the information system "Automated information system of the enterprise for the production of cabinet furniture" (AIS "Case Furniture").

Analysis of the subject area was performed; the functional scheme of AIS "Enclosure Furniture" was developed; the concept was developed, the construction architecture and the system implementation platform were chosen; a conceptual model of the AIS "Enclosure Furniture" was designed; a logical model of the AIS system "Enclosure Furniture" was designed based on a conceptual model; the physical structure of the database server is defined.

The result of the work was AIS "Furniture Furniture", which provides the ability to create an order and plan for cutting a sheet of material, maintain documents on the performance of work and settlement with the customer, search for data according to the criteria set by the user, and also provides the ability to save and print the created documents, including the plan for cutting .

ЗМІСТ

ВСТУП.....	3
1. АНАЛІЗ ТЕХНОЛОГІЧНОГО ПРОЦЕСУ ВИГОТОВЛЕННЯ МЕБЛІВ.....	5
1.1 Функції та організаційна структура підприємств, які виготовляють меблі.....	5
1.2 Опис потоків даних.....	6
1.2.1 Моделювання процесів.....	6
1.2.2 Діаграма потоків даних.....	13
1.3 Висновки до розділу.....	15
2. АНАЛІЗ ІСНУЮЧИХ САПР ВИГОТОВЛЕННЯ МЕБЛІВ.....	16
2.1 Огляд існуючих технічних рішень та виявлення їх переваг і недоліків.....	16
2.2 Обґрунтування необхідності розробки інформаційної системи...	17
2.3 Вимоги до системи.....	18
2.3.1 Вимоги до системи в цілому.....	18
2.3.2 Вимоги до функцій, що виконує система.....	21
2.3.3 Вимоги до виду забезпечення.....	22
2.4 Етапи розробки автоматизованої інформаційної системи.....	23
2.5 Висновки до розділу.....	24
3. ПРОЕКТУВАННЯ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ АІС ВИГОТОВЛЕННЯ КОРПУСНИХ МЕБЛІВ.....	25
3.1 Розробка архітектури побудови та платформи реалізації АІС....	25
3.1.1 Архітектура побудови.....	25
3.1.2 Структура та склад функціональних підсистем.....	27
3.1.3 Технічне забезпечення.....	30
3.2 Інформаційне забезпечення.....	31
3.2.1 Опис моделі інформаційної бази.....	31
3.2.2 Опис логічної структури інформаційної бази.....	34
3.3 Висновки до розділу.....	36

4. ФІЗИЧНА ТА ПРОГРАМНА РЕАЛІЗАЦІЯ АІС ВИГОТОВЛЕННЯ КОРПУСНИХ МЕБЛІВ.....	37
4.1 Опис фізичної реалізації бази даних.....	37
4.2 Опис структури програмного забезпечення.....	40
4.3 Алгоритмізація типових інформаційних запитів.....	43
4.3.1 Алгоритм реалізації розкрою.....	43
4.3.2 Типові інформаційні запити.....	45
4.4 Висновки до розділу.....	51
ВИСНОВКИ.....	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	53

ВСТУП

Актуальність теми. В даний час у людей існують великі потреби у предметах меблів, що виготовлені спеціально для інтер'єру або мають відповідну функціональність. Це пов'язано з тим, що стандартизовані предмети меблів, які масово існують на ринку, не завжди мають потрібний розмір, вигляд та функціонал.

Велика кількість несистематизованої інформації, що зберігається в паперовому варіанті, ускладнює своєчасне і точне виконання замовлень, є складність і в зберіганні інформації про замовників та про виконанні замовлення.

Оскільки дані зберігаються в паперовому варіанті або у вигляді окремих файлів Excel і Word, то при розрахунках з клієнтами часто відбувається втрата інформації про виконання замовлення, тобто є ризик не виконати замовлення через відсутність даних про замовника.

Розрахунок вартості замовлення ведеться вручну. У вартість замовлення включається вартість матеріалів, робіт з виготовлення та доставки.

На підставі перерахованих вище факторів було прийнято рішення розробити проект інформаційної системи «Автоматизована інформаційна система підприємства з виготовлення корпусних меблів» (AIC «Корпусні меблі»).

Основне призначення AIC «Корпусні меблі» - автоматизація роботи підприємства, що виготовляє корпусні меблі.

Загострення конкуренції серед промислових підприємств і необхідність зниження собівартості виробленої продукції вимагають підвищення ефективності виробництва, більш раціонального витрачання наявних в його розпорядженні фінансових і матеріальних засобів та ресурсів.

Підвищення ефективності виробництва може досягатися за рахунок оптимізації виробничих процесів, в результаті чого скорочується час простою обладнання, досягається значна економія сировини, підвищуються обсяги і

якість продукції, що випускається при аналогічних трудових і виробничих витратах.

Для вирішення завдань оптимізації виробничих процесів можна ефективно використовувати методи математичного моделювання та оптимізації із застосуванням комп'ютерних технологій.

З огляду на зазначені причини, актуальним є створення АІС «Корпусні меблі» є зниження економічних витрат на матеріали за рахунок оптимізації їх використання.

Об'єкт дослідження – технологічний процес виготовлення корпусних меблів.

Предмет дослідження – процес розкрою листа матеріалу, згідно заданих характеристик деталей.

Мета і завдання дослідження – зниження економічних витрат на матеріали за рахунок оптимізації їх використання.

Для досягнення мети роботи поставлені й розв'язані наступні задачі: обґрунтовано необхідність розробки інформаційної системи; розроблено інформаційне забезпечення системи; виконано опис фізичної реалізації бази даних.

Апробація результатів досліджень. Участь в LVII науково-практичній конференції ЦНТУ до «Дня науки-2023» (20 квітня 2023 року, м. Кропивницький). Тема доповіді: "Функціональна схема автоматизованої інформаційної системи «Корпусні Меблі»".

Структура та обсяг роботи. Бакалаврська робота складається із вступу, чотирьох розділів, висновків та списку використаних джерел.

1 АНАЛІЗ ТЕХНОЛОГІЧНОГО ПРОЦЕСУ ВИГОТОВЛЕННЯ МЕБЛІВ

1.1 Функції та організаційна структура підприємств, які виготовляють меблі

Як правило, підприємства, які займаються виготовленням меблів (в тому числі приватні підприємства) мають в своєму складі адміністративний відділ (складається з бухгалтера і менеджера), виробничий відділ (майстер-технолог і робітники) і відділ доставки (кур'єр).

Майстер-технолог займається роботою з ескізами і проектами і здійснює керівництво трьома робітниками, які виконують безпосереднє виготовлення меблів. Також майстер-технолог на підставі розробленого проекту готує бухгалтеру список матеріалів, які необхідно закупити.

В задачі адміністративного відділу, що складається з бухгалтера і менеджера, входить приймання та реєстрація замовлення з подальшим розрахунком з клієнтом. Крім того, в обов'язки бухгалтера входить закупівля матеріалів у сторонніх фірм в кількості, зазначеній майстром-технологом.

Коли замовлення готове, кур'єр отримує у бухгалтера рахунок, спільно з одним з робітників прибуває до замовника в попередньо обговорене час, потім робітник збирає замовлення і приймає оплату від замовника у відповідності з отриманим від бухгалтера рахунком. Замовник підписує акт про виконані роботи.

Після доставки замовлення кур'єр здає документи і гроші бухгалтеру, який контролює правильність розрахунків і оформлення документа. Потім менеджер закриває заявку в списку поточних замовлень.

Побудову структури підприємства можна розбити на три етапи:

- побудова організаційної моделі;
- побудова функціональної моделі;
- побудова інформаційної моделі.

Організаційна структура підприємства відображена на рис. 1.1.

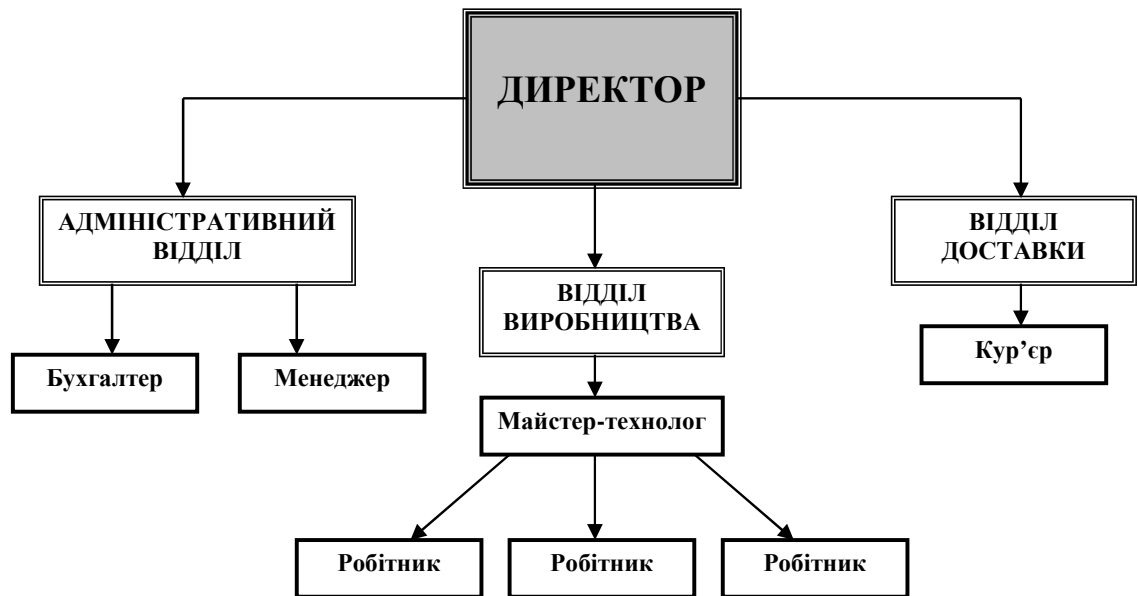


Рисунок 1.1 – Організаційна модель підприємства

Директор здійснює загальне керівництво виробничо-господарською і фінансово-економічною діяльністю підприємства, організовує взаємодію всіх його структурних підрозділів.

Відділ виробництва здійснює розробку проекту виробу, його подальше виготовлення та складання.

Адміністративний відділ, що складається з бухгалтера і менеджера, приймає та реєструє замовлення з подальшим розрахунком з клієнтом.

Відділ доставки здійснює транспортування частин виробів, підготовлених до збірки, за адресою, вказаною замовником.

1.2 Опис потоків даних

1.2.1 Моделювання процесів

Бізнес-процес - це сукупність взаємопов'язаних заходів або завдань, спрямованих на створення певного продукту або послуги для споживачів. Для наочності бізнес-процеси візуалізують за допомогою блок-схеми бізнес-процесів [1].

Моделювання бізнес-процесів - діяльність по формуванню моделей організацій, що включає опис ділових об'єктів (підрозділів, посад, ресурсів, ролей, процесів, операцій, інформаційних систем, носіїв інформації тощо) і зв'язків між ними.

Вимоги до сформованих моделей і їх відповідний зміст визначаються цілями моделювання. Бізнес-моделюванням також називають дисципліну і окремих підпроцес в процесі розробки програмного забезпечення, в якому описується діяльність підприємства і визначаються вимоги до системи - ті підпроцеси і операції, які підлягають автоматизації в інформаційній системі [2].

Проаналізувавши діяльність меблевого цеху, і провівши перед-проектне дослідження, можна виділити три основних процеси:

1. Прийняття замовлення.
2. Виготовлення частин виробу.
3. Розрахунок з замовником.

Функціональне моделювання бізнес-процесів представлено методологією IDEF0 (Integrated DEFinition). Вона описує ті ділові процеси, які протікають в об'єкті автоматизації. Основу методології IDEF0 складає графічна мова опису бізнес-процесів. Модель в IDEF0 представлена сукупністю ієрархічно впорядкованих і логічно пов'язаних діаграм. Кожна діаграма розташовується на окремому аркуші. Можна виділити чотири типи діаграм:

- контекстну діаграму A-0 (в кожній моделі може бути тільки одна контекстна діаграма);
- діаграми декомпозиції (в тому числі діаграма першого рівня декомпозиції A0, яка розкриває контекстну);
- діаграми дерева вузлів;
- діаграми тільки для експозиції (FEO).

Контекстна діаграма є вершиною деревовидної структури діаграм і являє собою загальний опис системи та її взаємодії із зовнішнім середовищем (як правило, тут описується основне призначення об'єкта, що моделюється).

Після опису системи в цілому, проводиться розбиття її на великі

фрагменти. Цей процес називається функціональною декомпозицією, а діаграми, які описують кожен фрагмент і взаємодію фрагментів, називаються діаграмами декомпозиції. Після декомпозиції контекстної діаграми (тобто, отримання діаграми A0) проводиться декомпозиція кожного блоку діаграми A0 на більш дрібні фрагменти і так далі, до досягнення потрібного рівня опису. Після кожного сеансу декомпозиції проводяться сеанси експертизи - експерти предметної області (зазвичай це опитування аналітиків співробітників підприємства) вказують на відповідність реальних бізнес-процесів створеним діаграм. Знайдені невідповідності виправляються, і тільки після проходження експертизи без зауважень можна приступати до наступного сеансу декомпозиції. Так досягається відповідність моделі реальним бізнес-процесам на кожному рівні моделі. Синтаксис опису системи в цілому і кожного її фрагмента однаковий у всій моделі [3].

Головна бізнес-функція - виготовлення меблів. Вхідними даними є заявка на виконання робіт. Вихідними - акт про виконані роботи. В якості управління виступають:

- порядок обслуговування клієнта;
- ДСТУ та ТУ по виготовленню меблів;
- проекти готових рішень;
- характеристики матеріалів;
- методики виготовлення.

Інструментами виконання головної бізнес-функції служать співробітники підприємства (бухгалтер, менеджер, кур'єр, робітники, майстер-технолог), а також обладнання, необхідне для виготовлення меблів. За допомогою програмного пакету ERwin Process Modeler було побудовано діаграму IDEF0 для головної бізнес-функції, яка представлена на рис. 1.2.

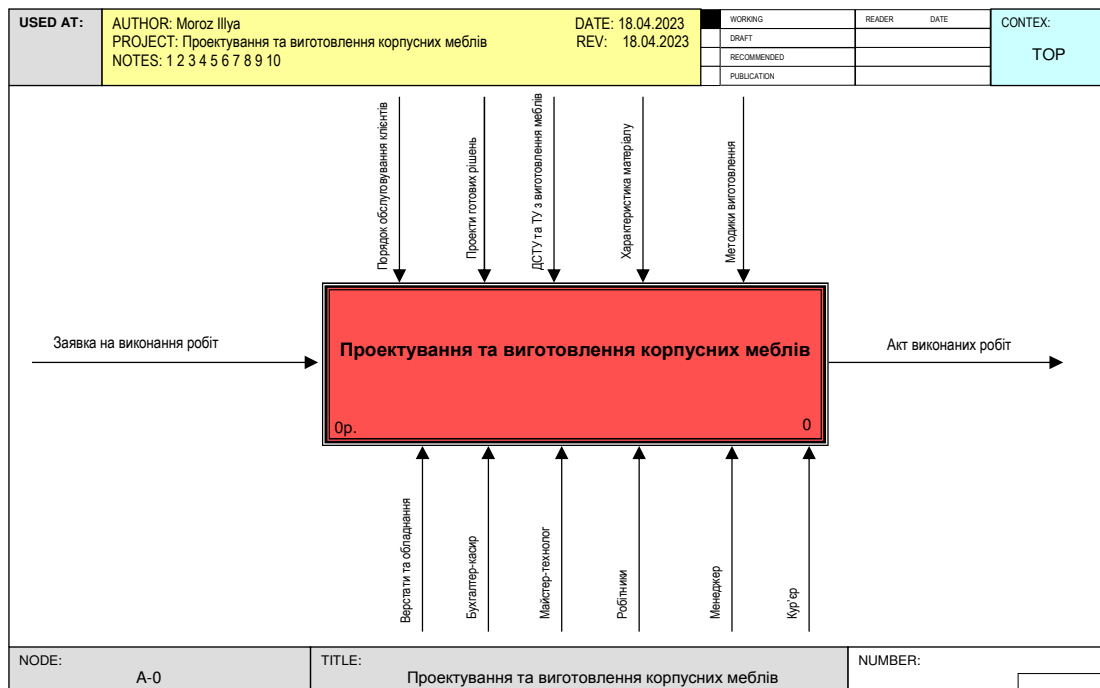


Рисунок 1.2 – Контекстна діаграма

Відповідно до зазначених вище, трьома основними бізнес-процесами при декомпозиції головної бізнес-функції є наступні функціональні блоки:

- прийняття замовлення;
- виготовлення частин виробу;
- розрахунок із замовником.

Діаграма декомпозиції контекстної діаграми представлена на рис. 1.3.

Функціональний блок «Прийняття замовлення» відповідає за реєстрацію замовлення, розробку ескізу, проекту виробу, узгодження з замовником.

У разі внесення замовником будь-яких змін, ескіз та проект розробляються заново. В управлінні блоком «Прийняття замовлення» беруть участь характеристики матеріалів, порядок обслуговування клієнтів, проекти готових рішень, ДСТУ з виготовлення меблів. На вхід надходить заявка на виконання робіт, на виході отримано узгоджений проект та договір. У якості виконавців виступають менеджер та майстер-технолог. Діаграма декомпозиції блоку «Прийняття замовлення» представлена на рис. 1.4.

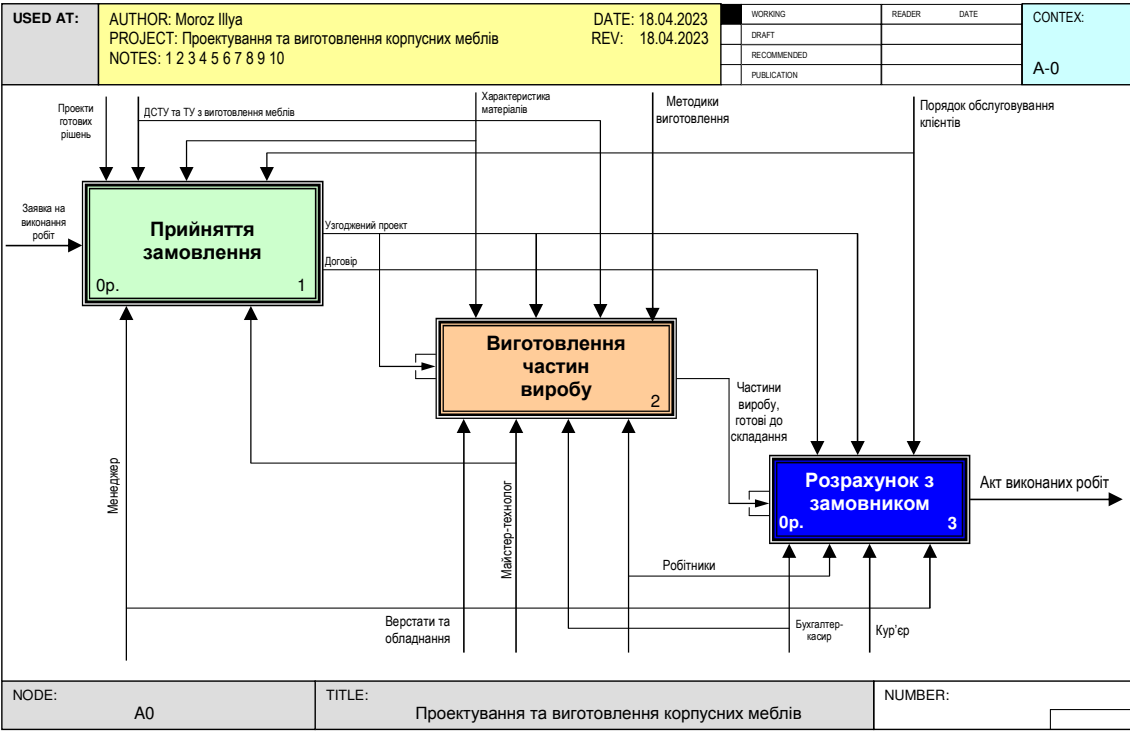


Рисунок 1.3 – Діаграма декомпозиції контекстної діаграми

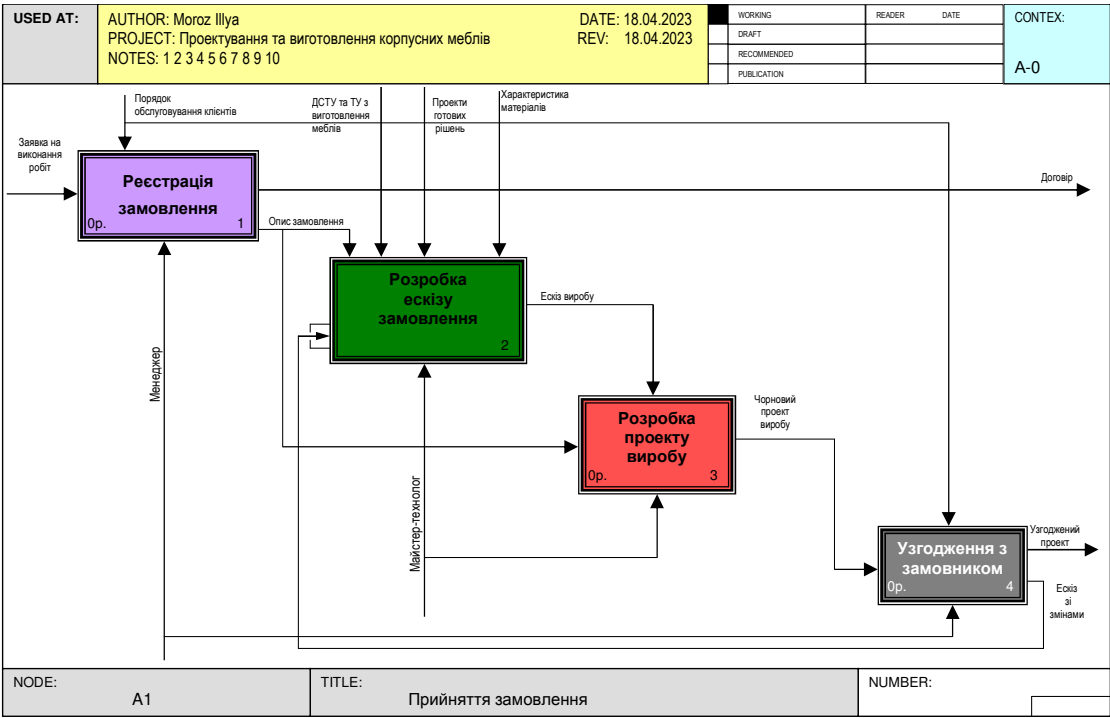


Рисунок 1.4 – Діаграма декомпозиції блоку

«Прийняття замовлення»

Функціональний блок «Виготовлення частин виробу» отримує в якості

вхідних даних узгоджений проект, в управлінні беруть участь характеристики матеріалів, методики виготовлення, ДСТУ з виготовлення меблів, а також узгоджений проект. У якості виконавців виступають майстер-технолог, робітники, верстати та обладнання. В якості вихідних даних виступають частини виробу, готові до складання.

Блок «Виготовлення частин виробу» має в своєму складі наступні блоки:

- замовлення і отримання матеріалів;
- розкрій плити на деталі;
- свердління сполучних отворів;
- облицювання кромки.

Діаграма декомпозиції блоку «Виготовлення частин виробу» представлена на рис. 1.5. Для автоматизації процесу розкрою потрібно уточнити, які процеси входять в блок «Розкрий плити на деталі». Тому була складена і його діаграма декомпозиції, яка представлена на рис. 1.6.

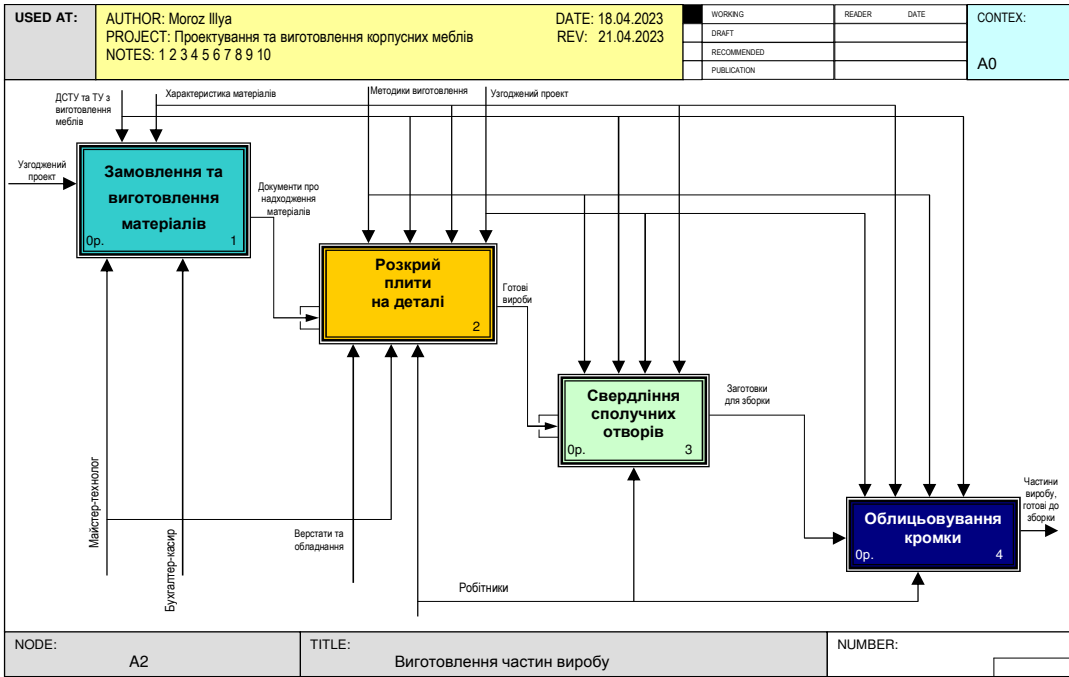
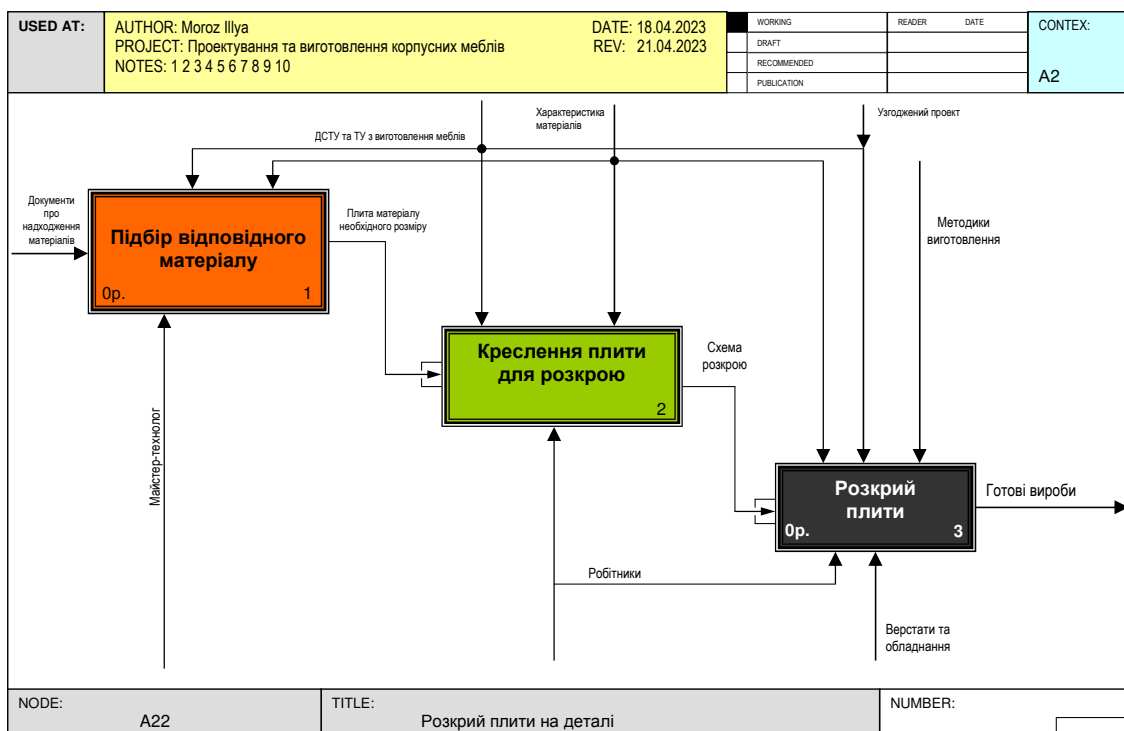


Рисунок 1.5 – Діаграма декомпозиції блоку «Виготовлення частин виробу»



**Рисунок 1.6 – Діаграма декомпозиції блоку
«Розкрій плити на деталі»**

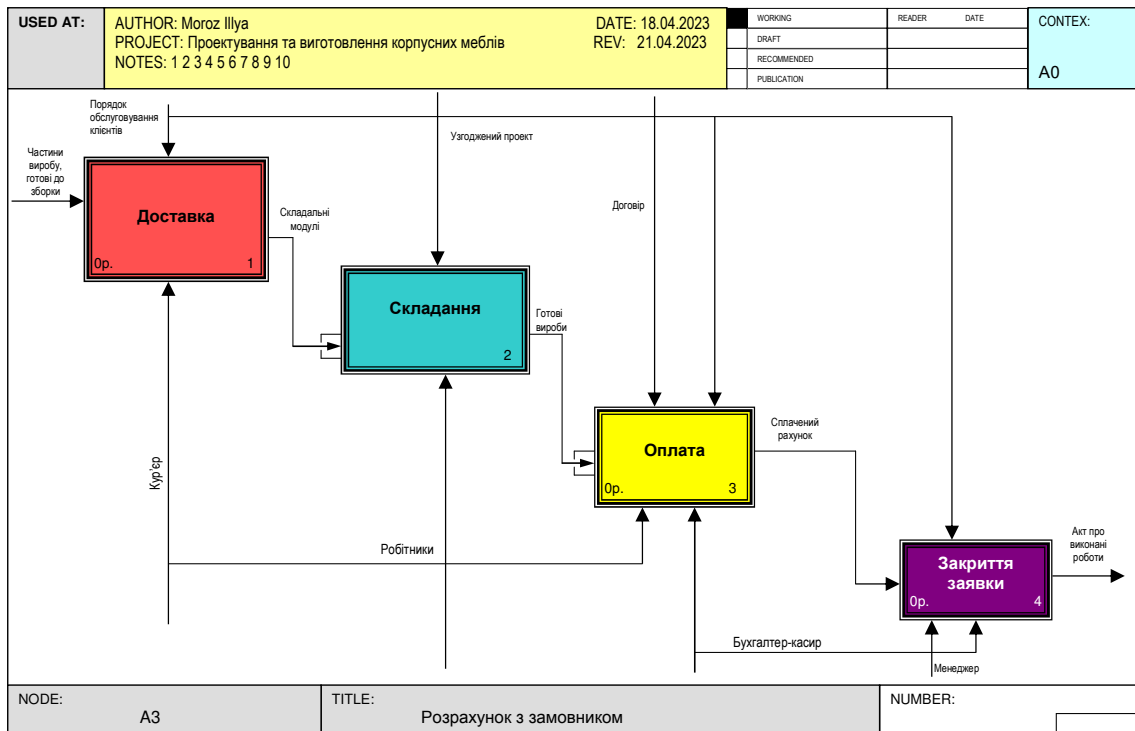
Функціональний блок «Розрахунок з замовником» включає доставку, збірку, оплату і закриття заявки.

На вхід надходять частини виробу, готові до складання, вихідними даними є акт про виконані роботи.

Керування цим блоком здійснюють: узгоджений проект; договір; порядок обслуговування клієнтів.

Виконавцями є робітники, кур'єр, бухгалтер-касир та менеджер.

Діаграма декомпозиції функціонального блоку «Розрахунок з замовником» представлена на рис. 1.7.



**Рисунок 1.7 – Діаграма декомпозиції блоку
«Розрахунок з замовником»**

1.2.2 Діаграма потоків даних

Діаграми потоків даних (Data Flow Diagramming) є основним засобом моделювання функціональних вимог до проектованої системи.

Вимоги представляються у вигляді ієрархії процесів, пов'язаних потоками даних. Діаграми потоків даних показують, як кожен процес перетворює свої вхідні дані у вихідні, і виявляють відносини між цими процесами [3, 4].

DFD-діаграми успішно використовуються як доповнення до моделі IDEF0 для опису документообігу та обробки інформації. Подібно IDEF0, DFD представляє систему, що моделюють, як мережу пов'язаних робіт.

Основні компоненти DFD - процеси або дії, зовнішні сутності, потоки даних, накопичувачі даних (сховища). На відміну від стрілок IDEF0, які представляють собою жорсткі взаємозв'язку, стрілки DFD показують, як об'єкти (включаючи дані) рухаються від однієї дії до іншої [3, 4].

На відміну від IDEF0, де система розглядається як пов'язані між собою дії, DFD розглядає систему як сукупність предметів. Контекстна діаграма часто включає дії і зовнішні посилання. Дії зазвичай іменуються за назвою системи, наприклад «Система обробки інформації».

Включення зовнішніх посилань в контекстну діаграму не скасовує вимоги методології чітко визначити мету, область і єдину точку зору щодо системи, яку моделюють.

У DFD дії (процеси) представляють собою функції системи, що перетворюють входи у виходи. Хоча дії зображуються прямокутниками із закругленими кутами, сенс їх збігається зі змістом дій IDEF0 і IDEF3. Так само, як процеси IDEF3, вони мають входи і виходи, але не підтримують управління та механізми, як IDEF0.

Зовнішні сутності зображують входи в систему і/або виходи із системи. Зовнішні сутності зображуються у вигляді прямокутника з тінню і зазвичай розташовуються по краях діаграми.

Одна зовнішня сутність може бути використана багаторазово на одній або декількох діаграмах. Зазвичай такий прийом використовують, щоб не малювати занадто довгих і заплутаних стрілок.

Потоки дій зображуються стрілками і описують рух об'єктів з однієї частини системи в іншу.

Оскільки в DFD кожна сторона дії не має чіткого призначення, як в IDEF0, стрілки можуть підходити і виходити з будь-якої грані прямокутника роботи.

У DFD також застосовуються двонаправлені стрілки для опису діалогів типу «команда-відповідь» між діями, між дією і зовнішньою сутністю і між зовнішніми сутностями.

На відміну від стрілок, що описують об'єкти в русі, сховища даних зображують об'єкти в спокої [4].

Діаграма потоків даних для бізнес-процесів представлена на рис. 1.8.

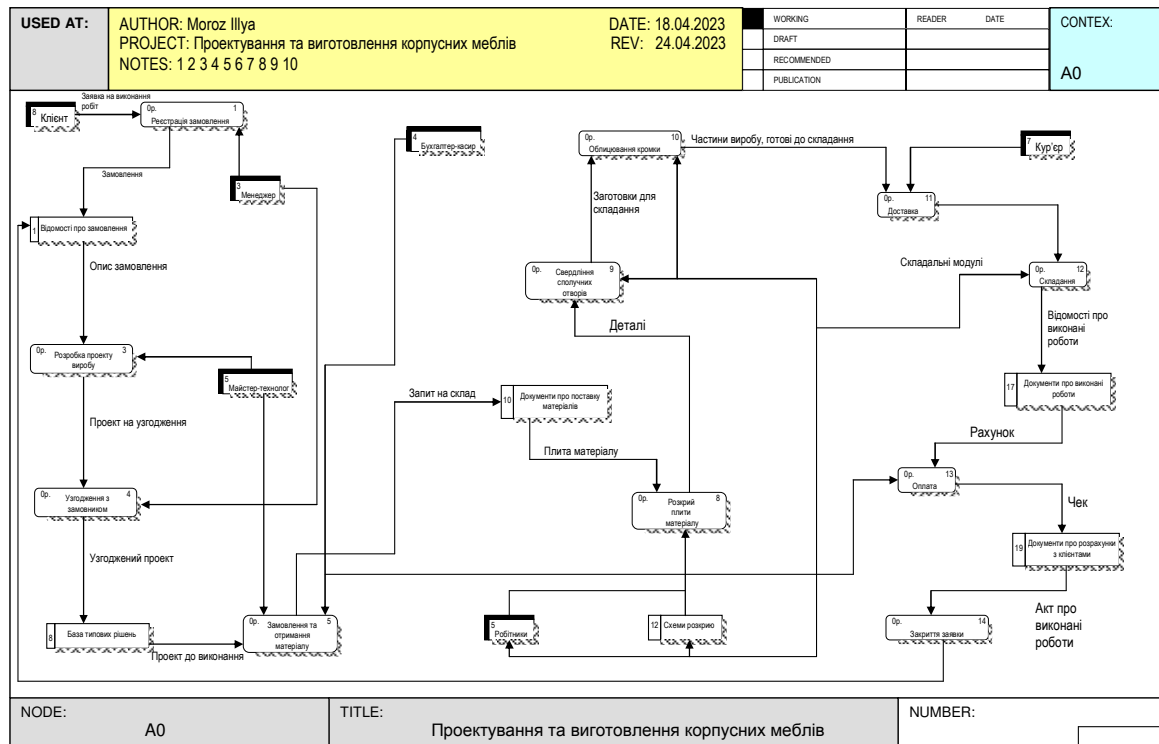


Рисунок 1.8 – Діаграма потоків даних

1.3 Висновки до розділу

В даному розділі виконано аналіз технологічного процесу виготовлення корпусних меблів. Підприємство має в своєму складі 3 відділи. Розглянуто та визначено функції і організаційну структуру підприємства.

Виконано опис даних, які необхідні для розробки інформаційної системи.

За допомогою методології IDEF0 було виконано моделювання бізнес-процесів підприємства. В результаті чого було побудовано контекстну діаграму та діаграми блоків «Прийняття замовлення», «Виготовлення частин виробу», «Розкрий плити на деталі», «Розрахунок із замовником».

Крім того, було побудовано загальну діаграму потоків даних (DFD) для бізнес-процесів підприємства.

2 АНАЛІЗ ІСНУЮЧИХ САПР ВИГОТОВЛЕННЯ МЕБЛІВ

2.1. Огляд існуючих технічних рішень та виявлення їх переваг і недоліків

Є безліч широковідомих програм, що використовуються для розкрою листа. Розглянемо деякі з них.

Розкрій Кузнєцова. За допомогою даної утиліти можна складати розкрій деревно-стружечної плити (ДСП), скла та будь-яких інших листових або рулонних матеріалів. Завдяки вбудованому потужному алгоритму прораховування, процедура створення розкрою відбувається непогано. За допомогою спеціального меню необхідно задавати потрібні дані для створення якісного розкрою. Програма призначена в першу чергу для професіоналів, тому новачкам буде досить складно з нею розібратися без попередньої підготовки. Утиліта Розкрій активно використовується великими підприємствами для прискорення і автоматизації процесу проектування. Додаток дозволяє об'єднувати замовлення, що виготовляються з одного матеріалу, тим самим, значно скорочуючи витрати і сприяє більш економічному розкрою при розрахунку.

Утиліта володіє потужними засобами для завдання всіляких умов. За допомогою даного програмного забезпечення можна задавати такі параметри як технологічний відхід, витрата матеріалу на розпил, відшліфовування краю, добавку додаткових аркушів і багато іншого [6].

Перевагами є: потужний спектр налаштувань; великий запас функціональності; автоматизація; зручний інтерфейс.

Недоліками є: програма платна; громіздкість; досить складна в освоєнні; обмеження демо-версії.

Астра-розкрій. Програма призначена для оптимізації розкрою листових матеріалів – деревно-стружкових плит, металу, скла і пластиків. Програма забезпечує введення і зберігання інформації про замовлення і матеріалах;

автоматичне і інтерактивне формування карт розкрою; розрахунок, збереження і облік відходів; друк карт розкрою і специфікацій [5].

Перевагами є: створення замовлення для розкрою; розкрий замовлення; редагування карти розкрою; друк технічної документації; розрахунок вартості замовлення і друк рахунку-фактури.

Недоліками є: висока вартість програми; громіздкий інтерфейс; нераціональне використання системних ресурсів комп'ютера.

2.2. Обґрунтування необхідності розробки інформаційної системи

На підприємстві реєстрація та облік замовлень виробляються вручну. Велика кількість несистематизованої інформації, що зберігається в паперовому варіанті, ускладнює своєчасне і точне виконання замовлень, є складність і в зберіганні інформації про замовників та про виконані замовлення.

Виготовлення замовлення так само пов'язане з деякими проблемами: труднощі при розкрої матеріалу і, як наслідок, нераціональна його витрата. Крім того, вже створені плани розкрою ніде не зберігаються, тому при надходженні замовлень, аналогічних вже виконаним, майстер-технолог змушений знову виконувати розрахунки і вирішувати, як буде розкроєна плита матеріалу.

Оскільки дані зберігаються в паперовому варіанті або у вигляді окремих файлів Excel і Word, то при розрахунках з клієнтами часто відбувається втрата інформації про виконання замовлення, тобто є ризик не виконати замовлення через відсутність даних про замовника.

Розрахунок вартості замовлення ведеться вручну. У вартість замовлення включається вартість матеріалів, робіт з виготовлення та доставки.

На підставі перерахованих вище факторів і з урахуванням можливостей існуючих проектних рішень, було складено список функцій, які необхідно реалізувати:

- створення замовлення;
- збереження замовлення;
- створення плану розкрою листа та можливість його збереження;
- друк плану розкрою, договори, рахунки;
- розрахунок суми замовлення;
- формування акту про виконані роботи.

2.3. Вимоги до системи

2.3.1. Вимоги до системи в цілому

Перелік підсистем, їх призначення та основні характеристики, вимоги до числа рівнів ієрархії і ступеня централізації системи.

Автоматизована інформаційна система (AIC) «Корпусні меблі» включає наступні підсистеми:

- Ухвалення замовлення;
- Виготовлення виробу;
- Розрахунок з замовником.

Підсистема «Ухвалення замовлення» призначена для реєстрації замовлення, розробки та погодження ескізу із замовником, а також подальшої розробки проекту виробу. Таким чином, замовлення спочатку реєструється менеджером, потім майстер-технолог розробляє ескіз, а потім і проект виробу, погоджує його з замовником і, при необхідності вносить корективи.

Підсистема «Виготовлення виробу» призначена для безпосереднього виготовлення частин виробу і включає в себе замовлення і отримання матеріалів і створення схеми розкрою плити на деталі.

Підсистема «Розрахунок з замовником» забезпечує доставку замовлення, складання частин виробу в меблі, оплату на основі сформованого рахунку і подальше закриття заявки з формуванням акта про виконані роботи.

Вимоги до способів і засобів зв'язку для інформаційного обміну між компонентами системи:

Інформаційний обмін здійснюється за допомогою локальної мережі.

Процес функціонування системи не передбачає взаємодії з іншими інформаційними системами.

Система повинна підтримувати багатокористувацький і автономний режими роботи. Користувачі здійснюють доступ до системи через локальну мережу.

Діагностика та оновлення системи виконують при необхідності.

Модернізація системи здійснюється на вимогу замовника системи.

В системі працюватимуть 4 співробітника: 1 співробітник відповідає за прийом і закриття заявок і замовлень, 1 займається створенням проекту виробу і його схем розкрою, 1 займається доставкою готових меблів, 1 займається розрахунками з замовниками.

Співробітник, що займаються прийомом заявок, повинен володіти навичками роботи з персональним комп'ютером на рівні користувача. Співробітник, який відповідає за розрахунками з замовниками, повинен володіти знаннями бухгалтерського обліку і звітності. Співробітник, який відповідає за створення схем розкрою, повинен володіти навичками володіння комп'ютером на рівні користувача, а також пройти навчання з використання системи «Корпусні меблі».

Чисельність персоналу може бути різною в залежності від обсягу і характеру робіт.

Система підтримує розкрій тільки на прямокутні деталі.

Виконання операцій повинно займати не більше 30 секунд.

Відновлення системи при помилках в роботі апаратних засобів (крім носіїв даних і програм) і помилки, пов'язаних з програмним забезпеченням, відновлення працездатності покладається на операційну систему (ОС).

При збоях в системі електропостачання апаратної частини, що призводять до перезавантаження ОС, відновлення програми має відбуватися після перезапуску ОС і запуску виконуваного файлу системи.

Також повинна бути забезпечена працездатність системи в цілому в разі

виникнення збоїв, аварій і відмов на окремих робочих станціях. Для захисту апаратури від кидків напруги і комутаційних перешкод повинні застосовуватися мережеві фільтри, а для можливості збереження даних користувачем при збої в системі електроживлення рекомендується застосовувати джерела безперебійного живлення.

Замовником забезпечується відповідність технічних рішень, використаних при модифікації і розробки підсистеми, чинним нормам, правилам техніки безпеки, пожежної безпеки і вибухобезпеки, охорони навколишнього середовища.

Умови експлуатації, а також види і періодичність обслуговування технічних засобів підсистеми повинні відповідати вимогам по експлуатації, технічного обслуговування, ремонту і зберігання, викладеним в документації заводу-виготовлювача (виробника) на них.

Системи обслуговує один працівник, який має восьмигодинний робочий день.

Система обслуговується щодня. Щотижня перевіряється наявність вірусів в операційній системі, проводиться оновлення вірусних баз і чищення реєстру. Дані функції виконує інженер-системотехнік.

Для захисту інформації від несанкціонованого доступу система повинна забезпечувати:

- а) ідентифікацію та аутентифікацію користувача;
- б) перевірку прав і обмежень доступу користувача на рівні функцій і масивів даних при роботі з системою.

Захист від впливу зовнішнього середовища повинна забезпечуватися засобами програмно-технічного комплексу Замовника.

Для даної системи повинна застосовуватися каскадна модель життєвого циклу програмного забезпечення (ПЗ).

В системі повинні використовуватися (при необхідності) загальні українські класифікатори і єдині класифікатори і словники для різних видів алфавітно-цифрової і текстової інформації.

Інтерфейс системи, файли довідок і будь-яка текстова інформація в програмі повинні бути російською або українською мовою.

Екранні форми повинні проектуватися з урахуванням вимог уніфікації:

Всі екранні форми користувальницького інтерфейсу повинні бути виконані в єдиному графічному дизайні, з однаковим розташуванням основних елементів керування та навігації.

2.3.2. Вимоги до функцій, що виконує система

Інформаційна система повинна забезпечувати виконання таких функцій:

- підсистема прийняття замовлення,
- підсистема виготовлення виробу,
- підсистема розрахунку з клієнтом.

Підсистема «Ухвалення замовлення» призначена для реєстрації замовлення, розробки та погодження ескізу із замовником, а також подальшої розробки проекту виробу. Таким чином, замовлення спочатку реєструється менеджером, потім майстер-технолог розробляє ескіз, а потім і проект виробу, погодить його з замовником і, при необхідності вносить корективи.

Підсистема «Виготовлення виробу» призначена для безпосереднього виготовлення частин виробу і включає в себе замовлення і отримання матеріалів і створення схеми розкрою плити на деталі.

Підсистема «Розрахунок з замовником» забезпечує доставку замовлення, складання частин виробу в меблі, оплату на основі сформованого рахунку і подальше закриття заявки з формуванням акта про виконані роботи.

Дані системи зберігаються на одній локальній машині. На вхід системи подається опис замовлення, на виході повинен бути рахунок і акт про виконані роботи.

Дані вводяться в систему вручну, обробляються і видаються користувачеві в необхідному вигляді (електронний, друкований).

У комплекс технічних засобів повинен входити джерело безперебійного живлення. Робота даного джерела повинна становити не менше півгодини для коректного завершення роботи системи.

Система повинна підтримувати автоматичне щоденне резервне копіювання.

Для вирішення завдання розкрою з мінімальними залишками буде використаний модифікований метод Нелдера- Міда (також відомий як метод деформованого багатогранника і симплекс-метод, - метод безумовної оптимізації функції від декількох змінних, що не використовує похідною (точніше - градієнтів) функції, а тому легко застосовний до негладких і / або зашумлених функцій. Суть методу полягає в послідовному переміщенні і деформації симплекса навколо точки екстремуму. Метод знаходить локальний екстремум і може «застрягти» в одному з них. Якщо все ж потрібно знайти глобальний екстремум, можна пробувати вибирати інший початковий симплекс) з використанням генетичного алгоритму[21] .

2.3.3. Вимоги до виду забезпечення

Система повинна працювати в операційних системах Windows / Vista.

У комплекс технічних засобів повинні входити такі елементи: робочі станції; джерело безперебійного живлення; середовище передачі даних між робочими станціями; принтер.

Технічні засоби купуються Замовником самостійно.

Процесор Intel Pentium IV 2 ГГц і вище, оперативна пам'ять не менше 2 Гб, обсяг жорсткого диска не менше 500 Гбайт.

Функціонування системи забезпечує інженер-системотехнік, в експлуатації беруть участь 4 співробітника.

Організаційне забезпечення повинно бути достатнім для ефективного виконання персоналом покладених на нього обов'язків при здійсненні функцій системи.

Захист від помилок персоналу полягає в перевірці заповнення даних в деяких полях, можливості відновлення вихідних даних і скасування останніх змін, розмежування доступу по функціям і повноваженням співробітників.

2.4. Етапи розробки автоматизованої інформаційної системи

Склад та зміст робіт для створення АІС представлені в табл. 2.1.

Таблиця 2.1 – Етапи розробки АІС «Корпусні меблі»

Стадії	Етапи робіт
1. Формування вимог до АІС	1.1. Дослідження об'єкту та необхідність створення АІС
	1.2. Формування вимог замовника до АІС
	1.3. Оформлення звіту про виконані роботи та заявки на розробку АІС
2. Розробка та затвердження технічного завдання	2.1. Розробка технічного завдання для створення АІС
	2.2. Затвердження технічного завдання для створення АІС
3. Розробка концепції АІС	3.1. Вивчення об'єкту АІС
	3.2. Проведення необхідних науково-дослідних робіт
	3.3. Розробка варіантів концепції АІС, що задовольняють вимоги користувача
4. Розробка АІС	4.1. Реалізація алгоритму розкрою
	4.2. Налаштування програми на тестових прикладах
	4.3. Реалізація структури даних у середовищі розробки
	4.4. Зв'язок структури даних з алгоритмом розрахунку
	4.5. Розробка програмного інтерфейсу
Стадії	Етапи робіт
5. Робоча документація	5.1. Розробка робочої документації та систему та її частини
	5.2. Адаптація програм
6. Введення в дію	6.1. Підготовка об'єкту автоматизації до введення АІС в дію
	6.2. Підготовка персоналу
	6.3. Комплектація АІС програмними засобами
	6.4. Проведення дослідної експлуатації і випробувань

2.5. Висновки до розділу

В даному розділі обґрунтовано необхідність створення автоматизованої інформаційної системи (АІС) з виготовлення корпусних меблів.

Наведено вимоги до системи та її програмного і апаратного забезпечення.

Визначено етапи розробки АІС для виготовлення корпусних меблів на підприємстві.

3 ПРОЕКТУВАННЯ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ АІС

ВИГОТОВЛЕННЯ КОРПУСНИХ МЕБЛІВ

3.1. Розробка архітектури побудови та платформи реалізації АІС

3.1.1. Архітектура побудови

Основними аспектами при виборі архітектури побудови ІС є швидкодія, надійність, масштабованість і безпека.

В даний час найбільш поширеними архітектурами є: файл-сервер; клієнт-сервер; багаторівнева архітектура.

У архітектурі файл-сервер сервер покладає на себе лише функцію зберігання даних, а обробка проводиться на клієнтських машинах. Це означає, що дані необхідно передавати по мережі, що призведе до сильного завантаженні мережевого трафіку. А це в свою чергу призведе до зниження продуктивності при збільшенні числа користувачів. Також при реалізації архітектури файл-сервер, проблема цілісності, узгодженості і одночасного доступу до даних вирішується децентралізовано: дані зберігаються на сервері, а обробляються на клієнті. Внаслідок цього знижується надійність програми. Ще одним недоліком є високі витрати на модернізацію і супровід сервісів бізнес-логіки на кожній клієнтській робочій станції. Однак дана архітектура має і низку переваг, таких як низька вартість розробки, висока швидкість розробки і невисока вартість оновлення та зміни програмного забезпечення.

Архітектура клієнт-сервер позбавлена недоліків вищеописаної архітектури, тому що сервер баз даних не тільки забезпечує доступ до загальних даних, але і виконує їх обробку. Клієнт посилає на сервер запити, на мові «зрозумілою» серверу, а він в свою чергу обробляє запит, контролюючи при цьому цілісність і узгодженість даних, і повертає на клієнт результат відпрацьованого запиту. В результаті навантаження на мережу знижується (клієнтові більше не потрібно обробляти проміжні дані). Зберігання та обробка проводиться централізовано, тому дана архітектура надійніше архітектури файл-сервер. До недоліків клієнт-серверної архітектури відносяться, по-перше, достатня складність розробки

системи через необхідність виконувати бізнес-логіку і забезпечувати інтерфейс з користувачем в одній програмі і високі вимоги до робочих станцій з тієї ж причини.

Наступною сходинкою розвитку архітектур ІС стала багаторівнева архітектура, в якій бізнес-логіка виконується на сервері додатків. Багаторівнева архітектура має наступні переваги: масштабованість; конфігурованість (ізолюваність рівнів один від одного дозволяє швидко і простими засобами переконфігурувати систему при виникненні збоїв або при плановому обслуговуванні на одному з рівнів); висока безпека; висока надійність; низькі вимоги до швидкості каналу (мережі) між терміналами і сервером додатків; низькі вимоги до продуктивності і технічним характеристикам терміналів, як наслідок зниження їх вартості.

Однак, незважаючи на незаперечні переваги, дана система не набула поширення, з наступних причин: складність розробки систем на основі багаторівневої архітектури (оскільки дуже складно «зістикувати» різні модулі, особливо якщо вони написані різними групами. А зміна в одному модулі, як правило, викликає лавиноподібні зміни в інших, і з цієї точки зору навіть просту систему, засновану на багаторівневій архітектурі, буде складніше виконати в 2 рази); високі вимоги до продуктивності серверів додатків і сервера бази даних, а, значить, і висока вартість серверного обладнання; високі вимоги до швидкості каналу (мережі) між сервером бази даних і серверами додатків; висока складність адміністрування [7].

Розглянувши всі переваги і недоліки кожної з архітектур, для реалізації системи АІС «Корпусні меблі» було обрано архітектуру клієнт-сервер. Дана архітектура дозволяє оптимально розподілити роботу між клієнтськими і серверними частинами системи (додаток, що працює на робочій станції, не читає записи бази даних «безпосередньо», а посилає запити на сервер, де вони послідовно обробляються, а результати обробки відсилаються на робочу станцію). А це суттєво скорочує інформаційні потоки в локальній обчислювальній мережі (ЛОМ).

Схема функціонування та побудови інформаційної системи представлена на рис. 3.1.

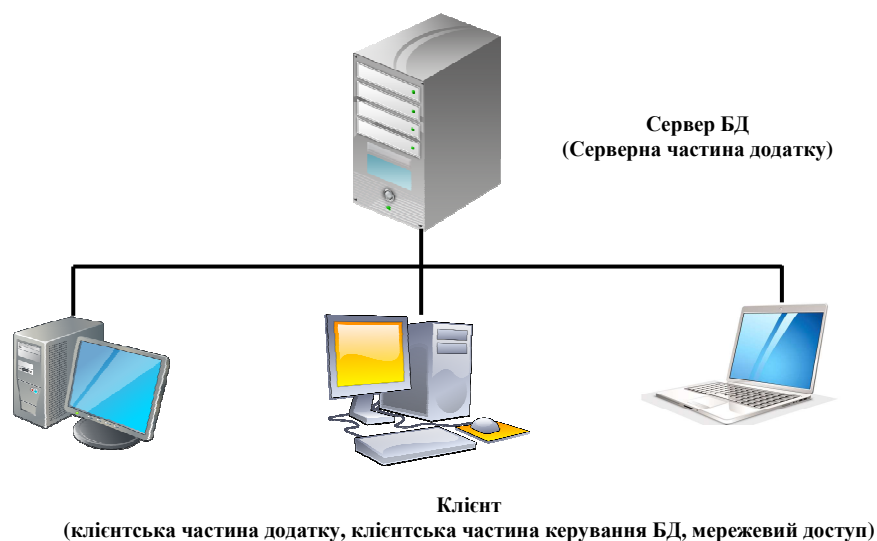


Рисунок 3.1 – Архітектура «Клієнт-Сервер»:
БД – база даних

3.1.2. Структура та склад функціональних підсистем

Функціональні підсистеми - комплекс економічних завдань з високим ступенем інформаційних обмінів (зв'язків) між завданнями (певний процес обробки інформації з чітко визначеною множиною вхідної і вихідної інформації. Наприклад, нарахування відрядної заробітної плати, облік приходу матеріалів, оформлення замовлення на закупівлю та інше.

Функціональні підсистеми інформаційно обслуговують певні види діяльності економічної системи (підприємства), характерні для його структурних підрозділів та (або) функцій управління. Інтеграція функціональних підсистем в єдину систему досягається за рахунок створення і функціонування підсистем, таких як:

- інформаційна;
- технічна;

- програмна;
- математична;
- лінгвістична.

Підсистеми, що забезпечують, є загальними для всієї ІС незалежно від конкретних функціональних підсистем, в яких застосовуються ті чи інші види забезпечення. В роботі підсистеми, що забезпечують, і організаційні підсистеми об'єднані в одну підсистему, що забезпечує. Обґрунтуванням такого рішення можна вважати, що їх складові забезпечують реалізацію цілей і функцій системи.

Склад підсистем не залежить від обраної предметної області і має [8-11]:

- функціональну структуру;
- інформаційне забезпечення;
- математичне (алгоритмічне і програмне) забезпечення;
- технічне забезпечення;
- організаційне забезпечення,

а на стадії розробки ІС мають додаткове забезпечення [9-12]:

- правове;
- лінгвістичне;
- технологічне;
- методологічне;
- інтерфейси із зовнішніми ІС.

Інформаційне забезпечення - це сукупність засобів і методів побудови інформаційної бази. Воно визначає способи і форми відображення стану об'єкта керування у вигляді даних всередині ІС, документів, графіків і сигналів поза ІС [10-12].

Математичне забезпечення складається з алгоритмічного і програмного.

Організаційне забезпечення - це сукупність засобів і методів організації виробництва та управління ними в умовах впровадження ІС [11].

Метою організаційного забезпечення є:

- вибір і постановка задач управління;

- аналіз системи управління та шляхів її удосконалення;
- розробка рішень щодо організації взаємодії ІС і персоналу;
- впровадження завдань управління.

Організаційне забезпечення включає в себе методики проведення робіт, вимоги до оформлення документів, посадові інструкції та інше.

Алгоритмічне забезпечення являє собою сукупність математичних методів, моделей і алгоритмів, що використовуються в системі для вирішення завдань і обробки інформації.

Структура інформаційної системи, склад функціональних підсистем і підсистем, що забезпечують, представлена на рис. 3.2.

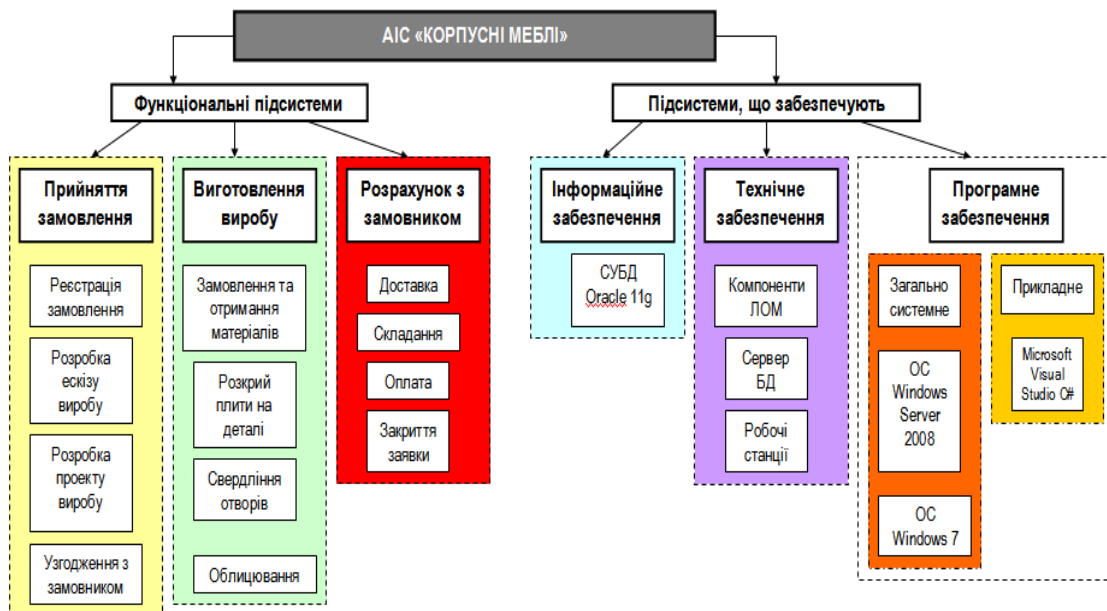


Рисунок 3.2 – Склад підсистем АІС «Корпусні меблі»:

СУБД – система управління базами даних;

ОС – операційна система;

ЛОМ – локальна обчислювальна мережа

3.1.3. Технічне забезпечення

В комплекс технічних засобів повинні входити такі елементи:

- робочі станції;

- джерела безперебійного живлення;
- кошти для побудови ЛВС;
- сервер БД;
- принтер.

Вимоги до сервера:

- пам'ять 8 Гб;
- процесор 2.2 ГГц Intel Xeon 5500 мінімум;
- швидкість диска SATA 8 Гбіт / с;
- мережевий адаптер 10 Гбіт / с;
- операційна система Windows Server 2008.

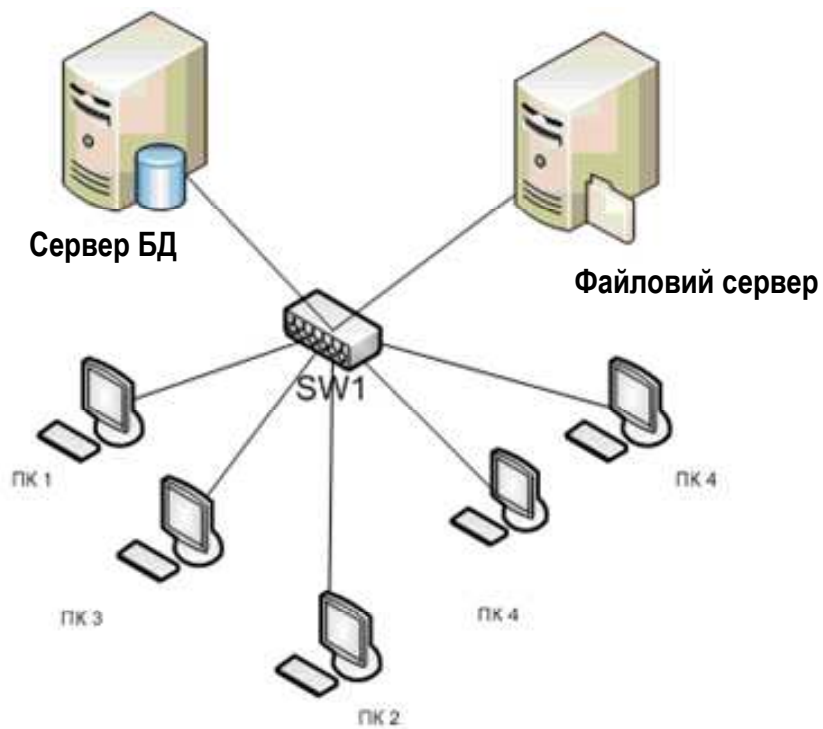
Вимоги до робочої станції:

- процесор 2 ГГц;
- пам'ять 2 Гб;
- жорсткий диск не менше 500;
- операційна система Windows 7;
- мережевий адаптер 100 Мбіт / с.

Технічні засоби ІС описані з урахуванням вимог до функціонування прикладного програмного комплексу. Технічні засоби повинні забезпечити:

- цілодобовий режим роботи комплексу технічних засобів і обладнання;
- гарантоване виконання всього комплексу програмного забезпечення в разі збою або виходу з ладу частини обладнання;
- захист даних від несанкціонованого доступу;
- сервера і робочі місця повинні бути об'єднані локальною мережею.

На рис. 3.3 представлена топологія локальної обчислювальної мережі (ЛОМ).



**Рисунок 3.3 – Логічна схема мережі підприємства
з виготовлення меблів:**

БД – бази даних; SW1 – комутатор;

ПК1,..., ПК4 – персональні комп'ютери

На даному рисунку видно, що через комутатор до сервера БД і файлового сервера підключено 5 робочих станцій. Топологія мережі - зірка.

3.2. Інформаційне забезпечення

3.2.1. Опис моделі інформаційної бази

Для зберігання даних в інформаційній системі використовується реляційна база даних під управлінням СУБД Oracle 11g. Концептуальна схема структури інформаційної бази наведена на рис. 3.4.

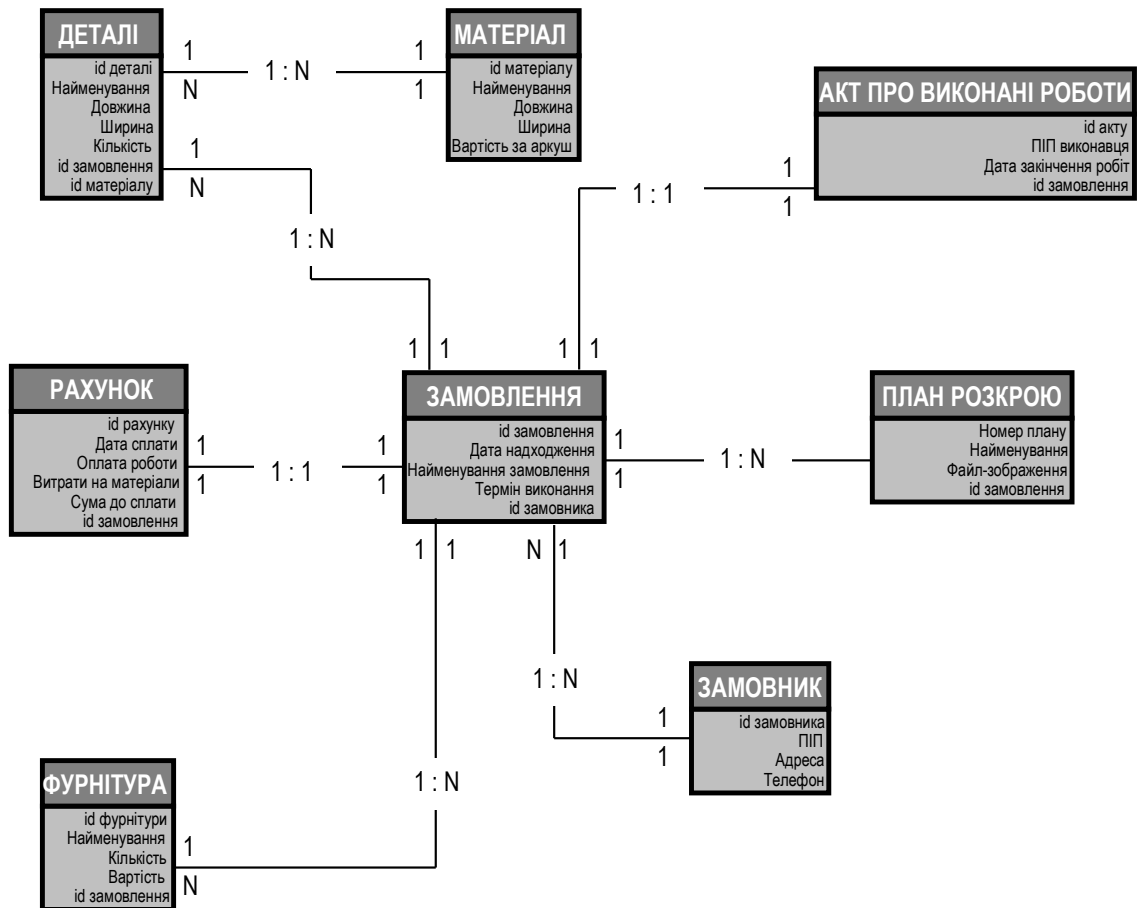


Рисунок 3.4 – Концептуальна схема бази даних

В АІС «Корпусні меблі» основними об'єктами є Замовлення, Деталі, Матеріали, Рахунок, Замовник, Фурнітура, План розкрою, Акт про виконані роботи. Розглянемо окремо кожен з об'єктів.

Сутність «Замовлення» має наступні атрибути: Номер замовлення, Дата надходження, Найменування замовлення, кількість комплектів, Термін виконання, Іd замовника, Іd плану.

Сутність «Деталі» має наступні атрибути: Іd деталі, Довжина, Ширина, Найменування, Іd замовлення, Іd матеріалу, Кількість.

Сутність «Матеріал» має атрибути: Найменування, Іd матеріалу, Ціна за лист, Довжина, Ширина.

Сутність «Рахунок» має атрибути: Іd рахунку, Сума до оплати, Іd замовлення.

Сутність «Замовник» має атрибути: Id замовника, ПІБ, адресу, телефон.

Сутність «Фурнітура» має атрибути: Id фурнітури, Найменування, Кількість, Вартість, Id замовлення.

Сутність «План розкрою» має атрибути: Id плану, Файл-зображення (змінна строкового типу, в якій буде зберігатися посилання на зображення з планом розкрою), Найменування.

Сутність «Акт про виконанні роботи» має атрибути: Id плану, Дата закінчення робіт, ПІБ виконавця, Id замовлення.

Між об'єктами Замовлення і Деталі максимальна потужність зв'язку 1:N, тобто одному замовленню може належати безліч деталей.

Між об'єктами Деталі й Матеріал максимальна потужність зв'язку 1:N, тобто на аркуші матеріалу може бути безліч деталей.

Між об'єктами Акт про виконанні роботи та Замовлення максимальна потужність зв'язку 1:1, тобто одному замовленню може відповідати 1 акт про виконання робіт.

Між об'єктами Замовлення і Рахунок максимальна потужність зв'язку 1:1, тобто замовленню може відповідати тільки 1 рахунок.

Між об'єктами Замовлення і План розкрою максимальна потужність зв'язку 1:N, тобто в замовленні може бути створено кілька планів розкрою.

Між об'єктами Замовлення і Замовник максимальна потужність зв'язку 1:N, тобто у одного замовника може бути безліч замовлень.

Між об'єктами Деталі й Матеріал максимальна потужність зв'язку 1:N, тобто на аркуші матеріалу може бути безліч деталей.

Між об'єктами Фурнітура та Замовлення максимальна потужність 1:N, тобто одному замовленню може відповідати безліч одиниць фурнітури.

3.2.2. Опис логічної структури інформаційної бази

Логічне (дatalogічне) проектування - створення схеми бази даних на основі конкретної моделі даних, наприклад, реляційної моделі даних. Для

реляційної моделі даних даталогічна модель - набір схем відносин, зазвичай із зазначенням первинних ключів, а також «зв'язків» між відносинами, що представляють собою зовнішні ключі.

Перетворення концептуальної моделі в логічну модель, як правило, здійснюється за формальними правилами. Цей етап може бути в значній мірі автоматизований [8].

Нормальна форма - якість зв'язку в реляційної моделі даних, що характеризує його з точки зору надмірності, потенційно призводить до логічно помилкових результатів вибірки або зміни даних. Нормальна форма визначається як сукупність вимог, яким має задовольняти відношення.

Процес перетворення відносин бази даних (БД) до виду, який відповідає нормальним формам, називається нормалізацією. Нормалізація призначена для приведення структури БД до виду, що забезпечує мінімальну логічну надмірність, і не має на меті зменшення або збільшення продуктивності роботи або ж зменшення (збільшення) фізичного обсягу бази даних. Кінцевою метою нормалізації є зменшення потенційної суперечливості збереженої в базі даних інформації. Загальне призначення процесу нормалізації полягає в наступному [9]:

- виключення деяких типів надмірності;
- усунення деяких аномалій оновлення;
- розробка проекту бази даних, який є досить «якісним» поданням реального світу, інтуїтивно зрозумілий і може служити основою для подальшого розширення;
- спрощення процедури застосування необхідних обмежень цілісності.

Усунення надмірності проводиться, як правило, за рахунок декомпозиції відносин таким чином, щоб в кожному відношенні зберігалися тільки первинні факти (факти, що не виводяться з інших збережених фактів) [9].

У створенні та розвитку теорії нормалізації брали участь багато вчених. Однак перші три нормальні форми і концепцію функціональної залежності запропонував Е. Кодд.

Перша нормальна форма (1NF): змінна відносини знаходиться в першій нормальній формі (1НФ) тоді і тільки тоді, коли в будь-якому допустимому значенні відносини кожен його кортеж містить тільки одне значення для кожного з атрибутів. У реляційної моделі відношення завжди знаходиться в першій нормальній формі за визначенням поняття відношення. Що ж стосується різних таблиць, то вони можуть не бути правильними уявленнями відносин і, відповідно, можуть не перебувати в 1НФ.

Друга нормальна форма (2NF): змінна відносини знаходиться в другій нормальній формі тоді і тільки тоді, коли вона знаходиться в першій нормальній формі і кожен неключовий атрибут функціонально повно залежить від її потенційного ключа.

Третя нормальна форма (3NF): змінна відносини знаходиться в третій нормальній формі тоді і тільки тоді, коли вона знаходиться в другій нормальній формі і відсутні транзитивні функціональні залежності неключових атрибутів від ключових.

Нормальна форма Бойса-Кодда (BCNF): змінна відносини знаходиться в нормальній формі Бойса-Кодда (інакше - в посиленій третій нормальній формі) тоді і тільки тоді, коли кожна її нетривіальна залежність та функціональна залежність, що приводиться ліворуч, має в якості свого детермінанта деякий потенційний ключ.

Четверта нормальна форма (4NF): змінна відносини знаходиться в четвертій нормальній формі, якщо вона знаходиться в нормальній формі Бойса - Кодда і не містить нетривіальних багатозначних залежностей.

П'ята нормальна форма (5NF): змінна відносини знаходиться в п'ятій нормальній формі (інакше - в проекційно-сполучній нормальній формі) тоді і тільки тоді, коли кожна нетривіальна залежність з'єднання в ній визначається потенційним ключем (ключами) цього відношення [10].

На логічному рівні виконується нормалізація бази даних, а також виділення ключів для кожної сутності. Логічні зв'язки реалізовані за допомогою первинних і зовнішніх ключів.

Логічна модель бази даних проектованої системи, в якій всі таблиці нормалізовані і виключені транзитивні залежності, представлена на рис. 3.5.

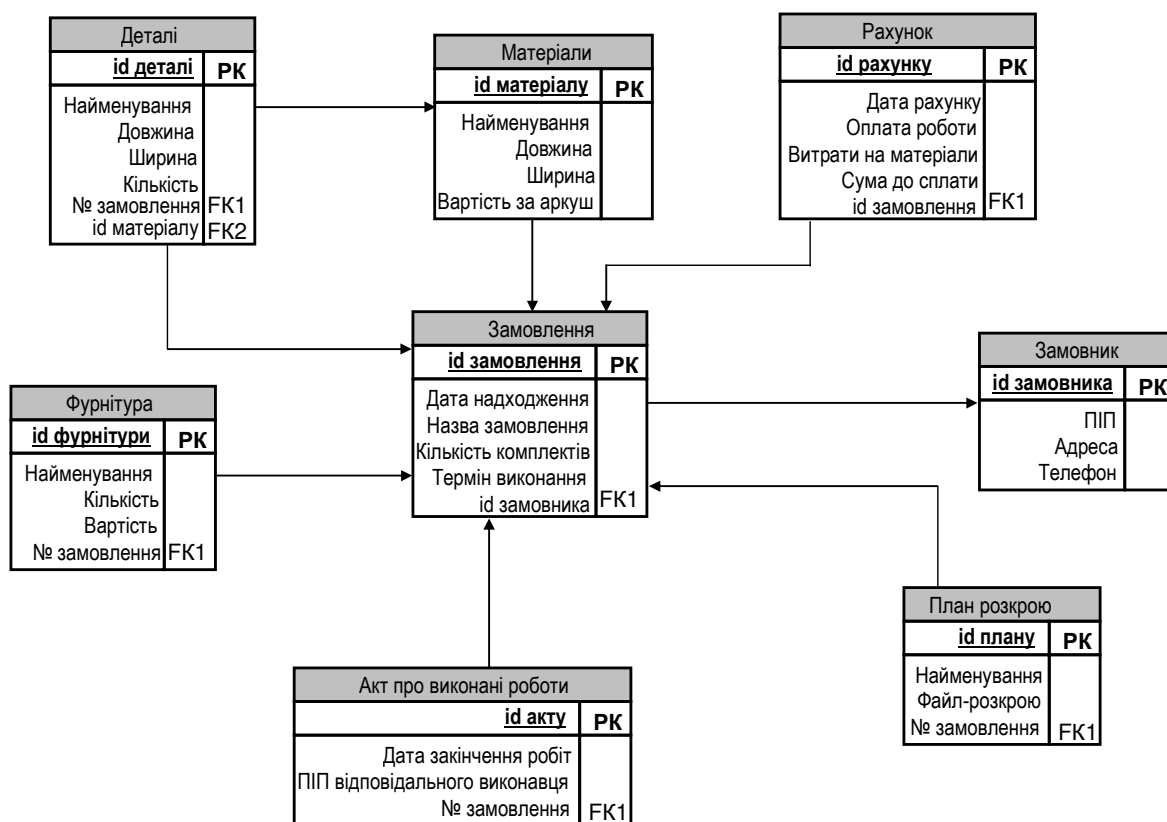


Рисунок 3.5 – Логічна структура інформаційної бази

3.3. Висновки до розділу

В даному розділі розглянуто основні архітектури побудови ІС, в результаті для подальшої роботи було обрано архітектуру «клієнт-сервер».

Наведено структуру та склад функціональних підсистем. Побудовано концептуальну та логічну моделі інформаційної бази.

4 ФІЗИЧНА ТА ПРОГРАМНА РЕАЛІЗАЦІЯ АІС ВИГОТОВЛЕННЯ КОРПУСНИХ МЕБЛІВ

4.1. Опис фізичної реалізації бази даних

Фізичне проектування - створення схеми бази даних для конкретної СУБД. Специфіка конкретної СУБД може включати в себе обмеження на іменування об'єктів бази даних, обмеження на підтримувані типи даних та інше. Крім того, специфіка конкретної СУБД при фізичному проектуванні включає вибір рішень, пов'язаних з фізичним середовищем зберігання даних (вибір методів управління дисковою пам'яттю, поділ БД по файлах і пристроях, методів доступу до даних), створення індексів та інше [11].

Фізична модель даних будується на базі логічної моделі і описує дані вже засобами конкретної СУБД. Відносини, розроблені на стадії логічного моделювання, перетворюються в таблиці, атрибути в стовпці, домени в типи даних, прийнятих в обраній конкретній СУБД.

Тобто у фізичній моделі між параметрами об'єкта і моделі однакової фізичної природи існує однозначна відповідність. У цьому випадку елементом системи ставляться у відповідність фізичні еквіваленти, які відтворюють структуру, основні властивості і співвідношення досліджуваного об'єкта. При фізичному моделюванні, основою якого є теорія подібності, зберігаються особливості проведення експерименту в натурі з дотриманням оптимального діапазону зміни відповідних фізичних параметрів [9].

У конструкторі були створені такі таблиці: Details, Materials, invoice, Furnitur, Order, Customer, Design_cutting, Akt_work_perfomed. Фізична модель АІС «Корпусні меблі» представлена на рис. 4.1.

Програма буде проектуватися по клієнт-серверній архітектурі з використанням модульного принципу проектування. В якості серверу баз даних буде виступати СУБД Oracle 11g. Застосування даної системи не вимагає витрат, оскільки використовується безкоштовна версія.

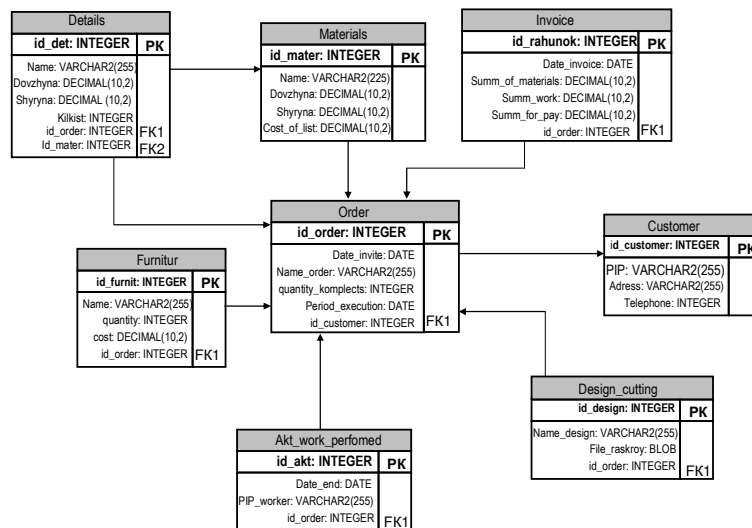


Рисунок 4.1 – Фізична модель

Опис створених таблиць бази даних наведені в табл. 4.1-4.8.

Таблиця 4.1 – Опис полів таблиці Order

Назва поля бази даних	Опис поля	Тип даних	Примітка
Id_order	Номер замовлення	Integer	PK
Data_invite	Дата замовлення	Date	-
Name_order	Ім'я замовлення	Varchar2(255)	-
quantity_komplekts	Кількість комплектів	Integer	-
Period_execution	Термін виконання	Date	-
Id_customer	Ідентифікаційний номер замовника	Integer	FK

Таблиця 4.2 – Опис полів таблиці Customer

Назва поля бази даних	Опис поля	Тип даних	Примітка
Id_customer	Ідентифікаційний номер замовника	Integer	PK
PIP	ППП замовника	Varchar2(255)	-
Adress	Адреса замовника	Varchar2(255)	-
Telephone	Телефон	Integer	-

Таблиця 4.3 – Опис полів таблиці Details

Назва поля бази даних	Опис поля	Тип даних	Примітка
Id_det	Номер деталі	Integer	PK
Name	Ім'я деталі	Varchar2(255)	-
Dovzhyna	Довжина деталі	Decimal(10,2)	-
Shyryna	Ширина деталі	Decimal(10,2)	-
Kilkist	Кількість таких деталей в проекті	Integer	-
Id_order	Ідентифікатор замовлення	Integer	FK
Id_mater	Ідентифікатор матеріалу	Integer	FK

Таблиця 4.4 – Опис полів таблиці Furnitur

Назва поля бази даних	Опис поля	Тип даних	Примітка
Id_furnit	Номер деталі	Integer	PK
Name	Ім'я деталі	Varchar2(255)	-
quantity	Кількість	Integer	-
cost	Вартість за штуку	Decimal(10,2)	-
Id_order	Ідентифікатор замовлення	Integer	FK

Таблиця 4.5 – Опис полів таблиці Materials

Назва поля бази даних	Опис поля	Тип даних	Примітка
Id_mater	Номер листа	Integer	PK
Name	Ім'я деталі	Varchar2(255)	-
Dovzhyna	Довжина	Decimal(10,2)	-
Cost_of_list	Вартість за аркуш	Decimal(10,2)	-
Shyryna	Ширина	Decimal(10,2)	-

Таблиця 4.6 – Опис полів таблиці Invoice

Назва поля бази даних	Опис поля	Тип даних	Примітка
Id_rahunok	Ідентифікаційний номер рахунку	Integer	PK
Date_invoice	Дата сплати	Date	-
Summ_for_pay	Сума до сплати	Decimal(10,2)	-
Summ_work	Оплата роботи	Decimal(10,2)	-
Summ_of_materials	Витрати на матеріали	Decimal(10,2)	-
Id_order	Номер замовлення	Integer	FK

Таблиця 4.7 – Опис полів таблиці Akt_work_performed

Назва поля бази даних	Опис поля	Тип даних	Примітка
Id_akt	Ідентифікаційний номер рахунка	Integer	PK
Data_end	Дата виконання	Date	-
PIP_worker	ППП виконавця	Varchar2(255)	-
Id_order	Номер замовлення	Integer	FK

Таблиця 4.8 – Опис полів таблиці Design_cutting

Назва поля бази даних	Опис поля	Тип даних	Примітка
Id_design	Ідентифікаційний номер плану	Integer	PK
File_raskroy	Файл з планом розкрою	Blob	-
Name_design	Найменування плану	Varchar2(255)	-
Id_order	Номер замовлення	Integer	FK

4.2. Опис структури програмного забезпечення

Oracle Database 11g допомагає зменшити витрати на інформаційні технології і підвищити якість послуг за рахунок консолідації в хмари баз даних і готові системи, наприклад, Oracle Exadata і Oracle Database Appliance. Це швидка, надійна, безпечна і легка в керуванні система для виконання всіх завдань, пов'язаних з базами даних, в тому числі з додатками для підприємств, банками даних і аналізом великих обсягів даних.

Oracle Exadata - це єдина машина баз даних, яка забезпечує найвищу продуктивність як для банків даних, так і онлайн-обробки транзакцій, що робить її ідеальною платформою для консолідації пов'язаних з базами даних змішаних навантажень в приватні хмарні середовища.

Oracle Database 11g пропонує економічно вигідне управління зберіганням даних завдяки автоматизації процесів, мінімізації дорогих операцій введення/виводу, стиску даних і максимальному використанню багаторівневої системи зберігання даних для всіх баз даних підприємства.

Починаючи з версії 10g компанія Oracle позиціонує свою СУБД як платформу для GRID обчислень. Концепція GRID обчислень досить проста, зрозуміла, гнучка і дозволяє економити кошти підприємства. Тому останнім часом спостерігається поступове впровадження цієї архітектури в IT інфраструктуру. Все більшої популярності набувають багатоядерні процесори.

Зараз більшість нових серверів має двох - чотирьох ядерні процесори і це не межа. Тому ера систем з тисячами процесорів вже не за горами. Потрібна платформа, що дозволяє ефективно реалізовувати програми на цій інфраструктурі. Oracle пропонує в якості такої платформи GRID на основі СУБД Oracle 11g.

Для реалізації клієнтської частини програми було застосовано середовище розробки Microsoft Visual Studio 2010. Microsoft Visual Studio - лінійка продуктів компанії Майкрософт, що включають інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів.

Дані продукти дозволяють розробляти як консольні додатки, так і додатки з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, підтримуваних Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework і Microsoft Silverlight.

Visual Studio включає в себе редактор вихідного коду з підтримкою технології IntelliSense і можливістю найпростішого рефакторінга коду. Вбудований відладчик може працювати як відладчик рівня вихідного коду, і як відладчик машинного рівня. Решта вбудованих інструментів включають в себе редактор форм для спрощення створення графічного інтерфейсу додатку, веб-редактор, дизайнер класів і дизайнер схеми бази даних.

Visual Studio дозволяє створювати і підключати сторонні додатки (плагіни) для розширення функціональності практично на кожному рівні, включаючи додавання підтримки систем контролю версій вихідного коду (наприклад, Subversion і Visual SourceSafe), додавання нових наборів інструментів (наприклад, для редагування і візуального проектування коду на предметно-

орієнтованих мовах програмування або інструментів для інших аспектів процесу розробки програмного забезпечення (наприклад, клієнт Team Explorer для роботи з Team Foundation Server) [12].

Було створено такі форми:

- KorpSbor.cs - екранна форма головного вікна програми;
- Dovidnyk.cs - екранна форма довідника матеріалів;
- Order.cs - екранна форма створення замовлення;
- Plan_raskroya.cs - екранна форма довідника планів розкрою;
- Plan_postr.cs - екранна форма діалогового вікна після побудови плану розкрою.

Також були створені екранні форми для довідників «Замовлення», «Деталі», «Фурнітура», «Замовники», «Акти», «Рахунки», їх інтерфейс аналогічний інтерфейсів довідників «Матеріали» і «Плани розкрою» і відрізняється лише іншими стовпцями даних.

4.3. Алгоритмізація типових інформаційних запитів

4.3.1. Алгоритм реалізації розкрою

Для побудови планів розкрою було використано модифікований метод Нелдера-Міда з використанням генетичного алгоритму.

Даний метод (далі - алгоритм) використовується для отримання щільної упаковки k прямокутників на кроці 2 генетичного алгоритму.

Дано k - число прямокутників. Для опису взаємного розташування прямокутників на площині було використано $2k$ -мірний простір. Координати центрів прямокутників це точка $2k$ -мірного простору.

Для застосування алгоритму Нелдера-Міда найважливішими кроками є вибір початкової точки і критерій перетворення симплекса. В алгоритмі, викладеному нижче, запропоновано новий спосіб вибору початкової точки, який дозволяє рухатися до вирішення не виходячи за межі побудованої

допустимої області в просторі розмірності $2k + 2$. Складність вибору критерію перетворення симплекса пов'язана з необхідністю запобігти передчасній зупинці алгоритму внаслідок потрапляння симплекса в локальний мінімум.

КРОК 1. Побудова правильного симплекса в просторі розмірності $2k + 2$. При побудові симплекса було використано властивість, що проекція будь-якої його вершини на протилежну грань збігається з центром мас цієї межі.

КРОК 2. Побудова обмежень на відстані між центрами прямокутників (кожного з кожним).

КРОК 3. Побудова обмежень для відстаней між центрами прямокутників і сторонами смуги (для кожного прямокутника з усіма сторонами). Ці обмеження було перетворено в обмеження простору $2k + 2$, як це було описано в попередньому кроці.

КРОК 4. Пошук максимальної і мінімальної висоти смуги.

КРОК 5. Обчислення поточної висоти смуги як середнього значення між максимальною і мінімальною висотою смуги.

КРОК 6. Вписуємо правильний симплекс в сферу радіусу. Збільшуємо розмір допустимої області шляхом зсуву обмежень на R .

КРОК 7. Обчислення цільової функції для кожної вершини симплексу. Цільова функція C - відстань від вершини до обмежень. За побудовою крапка не порушує обмеження, якщо значення цільової функції негативно (чим менше значення цільової функції, тим воно ближче до шуканого рішення). Було зменшено розмір допустимої області на величину, рівну модулю цільової функції, помноженому на коефіцієнт Q .

КРОК 8. Сортування вершин симплекса в залежності від значення цільової функції від максимальних до мінімальних значень.

КРОК 9. Відображення i -ї вершини симплексу щодо своєї протилежної грані. Було отримано точку I . Якщо значення цільової функції для точки I строго негативно і різниця між цільовою функцією і зрушенням менше ε (чим менше ε , тим точніше робота алгоритму), то рішення знайдено.

КРОК 10. Якщо значення цільової функції в точці I негативно та i -а

вершина симплексу не була одержана з точки I на попередньому перетворенні симплекса і виконано хоча б одна з таких умов 1 або 2, то тоді i -а вершина симплексу замінюється своєю протилежною точкою і здійснюється перехід на Крок 8.

КРОК 11. Якщо довжина ребра симплекса більше ϵ , то перехід на крок 7. В іншому випадку рішення не може бути знайдено і алгоритм завершує роботу.

Для більш точних результатів застосовують алгоритм ущільнення упаковки. В основі алгоритму ущільнення упаковки лежить метод лінійного програмування.

Крок 1. Вибираємо упаковку, генеруємо обмеження, що описують взаємне розташування всіх прямокутників відносно один одного всередині упаковки і кожного прямокутника щодо сторін смуги. Максимальною висотою буде отримана висота упаковки, а мінімальною - теоретично можлива.

Крок 2. Вибираємо поточну висоту як середнє значення між мінімальною і максимальною висотою.

Крок 3. Знаходимо точку всередині допустимої області.

Крок 4. Якщо допустима область існує, то максимальну висоту робимо поточній, інакше – мінімальну висоту робимо поточній. Якщо різниця між мінімальною і максимальною висотою більше деякого числа ϵ , переходимо на Крок 2, в іншому випадку алгоритм завершує свою роботу.

Слід також зауважити, що для збільшення швидкості роботи алгоритму можна збільшити допустиму область.

Для цього необхідно збільшити ширину смуги (наприклад, на 1-3%), а ширину прямокутників зменшити (на 1-2%). Далі алгоритм ущільнення упаковки, використовуючи реальні розміри смуги, прямокутників і дані про взаємне розташування прямокутників, перевіряє можливість даної упаковки. Якщо така упаковка можлива, то рішення знайдено. Якщо збільшення ширини смуги і зменшення ширини прямокутників призводять до неможливості реалізації такої упаковки, то упаковку необхідно проводити з вихідними даними [13].

4.3.2. Типові інформаційні запити

Інформаційні запити до бази в даній програмі будуть здійснюватися за допомогою мови структурованих запитів SQL.

SQL - універсальна комп'ютерна мова, яка застосовується для створення, модифікації та управління даними в реляційних базах даних. SQL ґрунтується на обчисленні кортежів [14].

Розглянемо деякі запити.

1. Вивід замовлення, згідно його номеру.

```
select *  
from Order  
where id_order = '111'
```

2. Вивід ППІ та адреси замовника, згідно дати замовника.

```
select zk.id_customer, zk.FIO, zk.Adress  
from Customer zk, Order zk  
where z.Date_invite = '25.05.2023'  
AND z.id_order = z.id_order
```

3. Вивід деталей по даті замовлення

```
select d.*  
from Detali d, Order z  
where z.Date_invite = '25.05.2023'  
AND d.id_order = z.id_order
```

Можливість виконання таких запитів буде включено в призначений для користувача інтерфейс.

Повний лістинг програми наведений нижче.

```
Текст модуля KorpSbor.cs  
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Windows.Forms;  
  
namespace КорпСбор_інтерфейсСшарп
```

```

{
    public partial class KorpSbor : Form
    {
        public KorpSbor()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
        }

        private void toolStripButton1_Click(object sender, EventArgs e)
        {
        }

        private void toolStripLabel1_Click(object sender, EventArgs e)
        {
        }

        e) private void створитиToolStripMenuItem1_Click(object sender, EventArgs
        {
        }

        private void toolStripTextBox1_Click(object sender, EventArgs e)
        {
        }

        private void toolStripMenuItem1_Click(object sender, EventArgs e)
        {
        }

        e) private void замовленняToolStripMenuItem1_Click(object sender, EventArgs
        {
        }

        private void створитиToolStripMenuItem_Click(object sender, EventArgs e)
        {
            Order f = new Order();
            f.Show(); // або так
        }

        e) private void матеріалиToolStripMenuItem_Click(object sender, EventArgs
        {
            Spravochnik Mater = new Dovidnyk();
            Mater.Show(); // або так
        }

        e) private void планиРозкроюToolStripMenuItem_Click(object sender, EventArgs
        {
            Plan_raskroya Plan = new Plan_raskroya();
            Plan.Show();
        }
    }
}

```

```

        private void відкритиДовідникToolStripMenuItem_Click(object sender,
EventArgs e)
        {

        }

        private void пошукПоНомеруToolStripMenuItem_Click(object sender, EventArgs
e)
        {
            PoshukOrderID Poshuk_O_ID = new PoshukOrderID();
            Poshuk_O_ID.Show();
        }

    }
}

```

Текст модуля PlanPostr.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace КорпСбор_інтерфейсСшарп
{
    public partial class Plan_postr : Form
    {
        public Plan_postr()
        {
            InitializeComponent();
        }

        private void pictureBox1_Click(object sender, EventArgs e)
        {

        }

    }
}

```

Текст модуля Order.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace КорпСбор_інтерфейсСшарп
{
    public partial class Order : Form
    {
        public Order()
        {
            InitializeComponent();
        }

        private void Form2_Load(object sender, EventArgs e)
        {

```

```

    }

    private void button1_Click(object sender, EventArgs e)
    {

    }

    private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
    {

    }

    private void label11_Click(object sender, EventArgs e)
    {

    }

    private void label4_Click(object sender, EventArgs e)
    {

    }

    private void dateTimePicker2_ValueChanged(object sender, EventArgs e)
    {

    }

    private void button7_Click(object sender, EventArgs e)
    {

    }

    private void label6_Click(object sender, EventArgs e)
    {

    }

    private void button3_Click(object sender, EventArgs e)
    {

        Plan_postr Plan_post = new Plan_postr();
        Plan_post.Show();

    }
}

```

Текст модуля Dovidnyk.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace КорпСбор_интерфейсСшарп
{
    public partial class Dovidnyk : Form
    {
        public Dovidnyk()
        {

```

```

        InitializeComponent();
    }

    private void Dovidnyk_Load(object sender, EventArgs e)
    {

    }

    private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
    {

    }

    private void bindingSource1_CurrentChanged(object sender, EventArgs e)
    {

    }
}
}

```

Текст модуля PlanRaskroya.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace КорпСбор_інтерфейсСшарп
{
    public partial class Plan_raskroya : Form
    {
        public Plan_raskroya()
        {
            InitializeComponent();
        }

        private void Plan_raskroya_Load(object sender, EventArgs e)
        {

        }
    }
}

```

Текст модуля PoshukOrderID.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace КорпСбор_інтерфейсСшарп
{
    public partial class PoshukOrderID : Form
    {
        public PoshukOrderID()
        {
            InitializeComponent();
        }
    }
}

```

```

    }

    private void PoshukOrderID_Load(object sender, EventArgs e)
    {

    }

}

```

Текст модуля Form1.cs

```

using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Form2 Pass = new Form2();
            Pass.Show();
        }
    }
}

```

Текст модуля Form2.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {

        }
    }
}

```

Наразі, триває робота над розробкою інтерфейсу користувача.

4.4. Висновки до розділу

В даному розділі наведено фізичну модель АІС «Корпусні меблі». В якості СУБД було обрано Oracle 11g. В результаті було створено форми для вікон програми. Для побудови планів розкрою було використано модифікований метод Нелдера-Міда разом з генетичним алгоритмом. Крім того, для більш точних результатів було використано алгоритм ущільнення упаковки.

ВИСНОВКИ

У відповідності з поставленими завданнями проектування інформаційної системи в рамках бакалаврської роботи було виконано наступне:

- аналіз предметної області;
- розроблено функціональну схему АІС «Корпусні меблі»;
- розроблено концепцію, обрано архітектуру побудови і платформу реалізації системи;
- спроектовано концептуальну модель АІС «Корпусні меблі»;
- спроектовано логічну модель системи АІС «Корпусні меблі» на основі концептуальної моделі;
- визначено фізичну структуру серверу баз даних.

Результатом роботи стала АІС «Корпусні меблі», що забезпечує можливість створення замовлення і плану розкрою листа матеріалу, ведення документів про виконання робіт і розрахунку з замовником, пошуку даних за заданими користувачем критеріями, а також забезпечує можливість збереження і друку створюваних документів, включаючи план розкрою.

Крім того, в АІС «Корпусні меблі» існує обмеження доступу користувача на рівні функцій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Функціональна модель системи. *CORE – Aggregating the world's open access research papers*. URL: <https://core.ac.uk/download/pdf/11320265.pdf>.
2. Федорчук В., Філіппова М. ВИКОРИСТАННЯ МЕТОДОЛОГІЇ IDEF0 В ПРОЕКТУВАННІ ВИРОБНИЧИХ ПРОЦЕСІВ. *Ефективність та автоматизація інженерних рішень у приладобудуванні* : Матеріали XVI Всеукр. науково-практ. конф. студентів, аспірантів та молодих вчен., м. Київ, 9 груд. 2020 р. 2020. С. 133–136. URL: <https://cutt.ly/871hoWb>.
3. Марк Девід А., Мак Гоуен, Клемент Л. «Методологія структурного аналізу і проектування» SADT, 1993. - с. 89.
4. Черемних С. В. «Структурний аналіз систем IDEF-технології» - Фінанси і статистика, 2001. - с. 147.
5. Програмне забезпечення для виробництва меблів. *Компанія Технос. Розробка програмного забезпечення для оптимізації розкрою промислових матеріалів та для виробництва меблів*. URL: <http://www.astrapro.com.ua/default.asp?page=astra-konstruktor-mebeli>.
6. Програми для проектування меблів онлайн - Blogchain. *Український Блог про Сучасні Інформаційні Технології Світу | IT - BLOG Blogchain*. URL: <https://blogchain.com.ua/prohramy-dlia-proektuvannia-mebliv-onlajn/>.
7. Тадео Мурата. «Мережі Петрі: властивості, аналіз, застосування» - 1989 – с. 77, с. 41-85.
8. Береза А.М. Основи створення інформаційних систем: Навчальний посібник. – К.: КНЕУ, 2001. – 214 с.
9. Дейт К. Дж. Введення до систем баз даних: Пер. з англ. / Дейт К. Дж. – 6-е видання. – Київ: Діалектика, 1998. – 784 с.
10. Ситник В.Ф., Писаревська Т.А., Єр'оміна Н.В., Краєва О.С. Основи інформаційних систем: Навчальний посібник. – К.: КНЕУ, 2001. – 420с.
11. Пономаренко В.С., Пушкар О.І., Коваленко Ю.І. Проектування автоматизованих економічних інформаційних систем: К.: ІЗМН, 1996. – 312с.

12. Рендольф Н. Visual Studio 2010 для професіоналів / Рендольф Н., Гарднер Д., Андерсон К., Минутілло М. - Київ: Діалектика, 2011.-1184 с.
13. Завадський І.О. Основи баз даних: [Навч. посіб.]/І.О. Завадський.Видавець: І.О. Завадський, 2011. — 192 с.
14. Голуб Б.Л., Ящук Д.Ю. Основи організації баз даних: навч. пос. К.: НУБіП, 2017. – 139 с.
15. Бардус І. О. Бази даних у схемах (на основі фундаменталізованого підходу) : навч. посіб. /І. О. Бардус, М. І. Лазарєв, А. О. Ніценко. – Харків : Вид-во «Діса плюс», 2017. – 133 с.
16. Системи бази даних і знань. Методичні вказівки до виконання практичних робіт студентами з напрямку підготовки 6.050201 “Системна інженерія” /Укл.: Д.В. Трушаков, М.О. Каліта, І.О. Скриннік. – Кіровоград: КНТУ, 2009. – 75 с.
17. Грицюк Ю. І. Моделювання карт і оптимізація плану розкрою плитних деревних матеріалів на меблевій заготовці / Український держ. лісотехнічний ун-т. — Л. : Панорама, 2004. — 524с.
18. Дячун З. Й. Конструювання меблів. Корпусні вироби: навч. посіб. для студ. вищ. навч. закл. / Національний лісотехнічний ун-т України. — К. : Видавничий дім «Києво-Могилянська академія», 2007. — 387с.
19. Розов В.І. Довідник майстра меблевого виробництва. - М.: Лісова промисловість, 2002. - С.-82.
20. Цвіркун А.Д., Акінфієв В.К. Аналіз інвестицій і бізнес-план: Методи та інструментальні засоби. - М.: Ось-89, 2002.
21. Учасники проєктів Вікімедіа. SQL – Вікіпідручник. *Вікіпідручник*. URL: <https://uk.wikibooks.org/wiki/SQL>.
22. Грицюк Ю. Оптимізація технологічного процесу розкрою плитних деревних матеріалів на меблевій заготовці. Основа, 2005. 484 с.