МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ім. Ю.М. Потебні
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ

КАФЕДРА ЕЛЕКТРОНІКИ, ІНФОРМАЦІЙНИХ СИСТЕМ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Кваліфікаційна робота

перший (бакалаврський)
(рівень вищої освіти)

на тему **Розробка веб-платформи для професійного обміну параметрами обробки та оптимізації процесів на CNC-верстатах**

Виконав: студент 4 курсу, групи 6.1212-ПЗС
спеціальності 121 Інженерія програмного забезпечення
(код і назва спеціальності)
освітньої програми Програмне забезпечення систем
(код і назва освітньої програми)
Я.А. Михайличенко
(ініціали та прізвище)
Керівник к.т.н., доцент, О.М. Міхайлуца
(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)
Рецензент директор ТОВ Дісітел
П.О. Лютий
(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Запоріжжя
2026

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ім. Ю.М. Потебні ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ

Кафедра електроніки, інформаційних систем та програмного забезпечення
Рівень вищої освіти перший (бакалаврський)
Спеціальність 121 Інженерія програмного забезпечення
(код та назва)
Освітня програма Програмне забезпечення систем
(код та назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри Тетяна КРИТСЬКА

“ 01 ” березня 2026 року

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

Михайличенку Ярославу Андрійовичу
(прізвище, ім’я, по батькові)

1. Тема роботи Розробка веб-платформи для професійного обміну параметрами обробки та оптимізації процесів на CNC-верстатах
керівник роботи Міхайлуца Олена Миколаївна, доцент,
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)
затверджена наказом ЗНУ від 23.12.2025 р. № 1830-с
2. Строк подання студентом кваліфікаційної роботи 02.06.2026
3. Вихідні дані бакалаврської роботи
   - комплект нормативних документів ;
   - технічне завдання до роботи.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
   - огляд та збір літератури стосовно теми кваліфікаційної роботи;
   - огляд та аналіз існуючих рішень та аналогів;
   - проєктування архітектури реляційної бази даних та Telegram-бота; розробка серверної частини застосунку, алгоритмів пошуку та системи модерації користувацького контенту;
   - створення програмного продукту та його опис;
   - перелік вимог для роботи програми;
   - розробка висновків та пропозицій.
5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень) слайдів презентації

6. Консультанти розділів бакалаврської роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата
		Завдання прийняв

7. Дата видачі завдання 01.03.2026

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів магістерської роботи	Примітка
1.	Аналіз предметної області.	02.03-04.03	виконано
2.	Формування вимог до системи та обґрунтування технологічного стеку.	05.03	виконано
3.	Проєктування архітектури застосунку та реляційної бази даних.	06.03-12.03	виконано
4.	Налаштування серверного оточення (Node.js, NestJS, Prisma).	13.03-18.03	виконано
5.	Розробка модуля роботи з Telegram Bot API (grammY).	19.03-26.03	виконано
6.	Проєктування внутрішнього формату-посередника.	27.03-05.04	виконано
7.	Розробка модулів управління каталогами матеріалів та інструментів.	05.04-16.04	виконано
8.	Створення підсистеми модерації користувацького контенту.	16.04-28.04	виконано
9.	Розробка системи соціальної взаємодії (рейтингування, коментування, збереження).	28.04-12.05	виконано
10.	Автоматизація наповнення бази даних еталонними режимами (Seed).	12.05-20.05	виконано
11.	Контейнеризація розробленого застосунку (Docker).	21.05-27.05	виконано
12.	Системне та функціональне тестування.	28.05-02.06	виконано
13.	Написання розділів пояснювальної записки.	02.06-04.06	виконано
14.	Оформлення презентації та графічного матеріалу.	05.06-09.06	виконано

Студент _____ _____Я.А. Михайличенко

Керівник роботи _____ _____О.М. Міхайлуца

Нормоконтроль пройдено

Нормоконтролер _____ _____І.А. Скрипник

АНОТАЦІЯ

Сторінок — 82

Рисунків — 18

Джерел — 17

Михайличенко Я.А. CNC Helper - Telegram-бот для пошуку режимів різання на верстатах з ЧПК: кваліфікаційна робота бакалавра спеціальності 121 «Інженерія програмного забезпечення» / наук. керівник О.М. Міхайлуца. Запоріжжя: ЗНУ, 2026. 82 с.

Робота присвячена розробці інформаційної системи для автоматизації пошуку та підбору технологічних параметрів обробки деталей на верстатах з числовим програмним керуванням (ЧПК).

Мета проєкту — створення доступного Telegram-бота, що дозволяє операторам швидко отримувати рекомендовані режими різання (швидкість шпинделя, подача, глибина) на основі комбінації матеріалу та інструменту.

Спроєктовано архітектуру бази даних для зберігання ієрархічних каталогів. Розроблено серверну частину на базі фреймворку NestJS із застосуванням Prisma ORM та PostgreSQL. Реалізовано багатокроковий скінченний автомат для навігації, систему управління доступом на основі ролей (RBAC), а також підсистему модерації користувацького контенту. Впроваджено соціальні механіки оцінювання (UPSERT-логіка) та збереження в «Обране».

Ключові слова: *верстати з ЧПК, режими різання, Telegram-бот, NestJS, Prisma ORM, PostgreSQL, grammY, краудсорсинг.*

ABSTRACT

Pages — 82

Drawings — 18

Sources — 17

Mykhailychenko Y.A. CNC Helper - Telegram bot for finding cutting modes on CNC machines: bachelor's thesis of specialty 121 "Software Engineering" / Supervisor O. M. Mikhailutsa Zaporizhzhia: ZNU, 2026. 82 p.

The thesis is devoted to the development of an information system for automating the search and selection of technological parameters for machining parts on computer numerical control (CNC) machines.

The goal of the project is to create an accessible Telegram bot that allows operators to quickly obtain recommended cutting modes based on the combination of material and tool.

A database architecture was designed to store hierarchical catalogs. The server side was developed using the NestJS framework, Prisma ORM, and PostgreSQL. A multi-step finite state machine for navigation, a Role-Based Access Control (RBAC) system, and a user content moderation subsystem were implemented. Social mechanics for evaluation (UPSERT logic) and saving to "Favorites" were introduced.

Keywords: *CNC machines, cutting modes, Telegram bot, NestJS, Prisma ORM, PostgreSQL, grammY, crowdsourcing.*

ЗМІСТ

ВСТУП 9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 15
1.1 Характеристика процесу обробки на верстатах з ЧПК 15
1.2 Огляд літературних джерел 18
1.3 Аналіз програмних продуктів-аналогів 22
1.3.1 CAM-системи провідних розробників 22
1.3.2 Онлайн-калькулятори виробників інструменту 25
1.3.3 Універсальні онлайн-калькулятори 27
1.3.4 Табличні довідники та довідкова література 29
1.3.5 Telegram-боти та месенджер-рішення 30
1.3.6 Порівняльна таблиця рішень 31
1.4 Постановка завдання 32
1.4.1 Мета розробки 32
1.4.2 Функціональні вимоги 32
1.4.3 Нефункціональні вимоги 33
1.4.4 Опис ролей користувачів 35
1.4.5 Структура даних 37
1.4.6 Технологічний стек 38
1.4.7 Завдання для реалізаці 38
2. ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ 40
2.1 Аналіз функціональних та нефункціональних вимог 40
2.1.1 Функціональні вимоги та ролі користувачів 40
2.1.2 Нефункціональні вимоги 41
2.2 Обґрунтування вибору технологічного стеку 42
2.2.1 TypeScript як мова розробки 42
2.2.2 Фреймворк NestJS та архітектурний патерн DI 42
2.2.3 Бібліотека grammY та Telegram Bot API 42
2.3 Проєктування структури бази даних 43

2.4 Проєктування функціоналу та моделі взаємодії 45
2.4.1 Архітектура взаємодії як скінченний автомат 45
2.4.2 Модель життєвого циклу контенту та модерація 46
2.4.3 Оптимізація представлення даних та пагінація 47
2.4.4 Механіка соціального рейтингу та UPSERT-логіка 47
2.5 Висновки до розділу 2 48
3. РОЗРОБКА ТА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ 50
3.1 Структура проєкту та налаштування середовища 50
3.2 Розробка модуля доступу до даних на основі Prisma ORM 52
3.2.1 Конфігурація схеми даних та генерація клієнта 52
3.2.2 Реалізація реляційних зв'язків та обмежень цілісності 52
3.2.3 Імплементація PrismaService в екосистему NestJS 53
3.2.4 Управління станом бази даних через міграції 54
3.3 Реалізація інтерфейсу користувача та бізнес-логіки Telegram-бота 54
3.3.1 Архітектура обробки повідомлень та Middleware 54
3.3.2 Механіка динамічних Inline-клавіатур 56
3.3.3 Реалізація покрокового сценарію пошуку режимів 57
3.3.4 Система соціальної взаємодії: рейтинги та коментарі 62
3.4 Реалізація підсистеми модерації та користувацького контенту 64
3.4.1 Механіка створення пропозицій (User-Generated Content) 64
3.4.2 Інтерфейс адміністратора та процес модерації 66
3.4.3 Персоналізований контроль та обробка винятків 68
3.4.4 Управління категоріями та елементами каталогу 70
3.5 Реалізація персоналізованого розділу «Обране» 70
3.6 Валідація даних та обробка виняткових ситуацій 71
3.6.1 Технічна валідація вхідних параметрів 71
3.6.2 Обробка логічних конфліктів та стану бази даних 73
3.6.3 Обробка помилок Telegram API 73
3.7 Автоматизоване наповнення бази даних (Seed-дані) 74
3.8 Тестування та розгортання системи 75

3.8.1 Функціональне тестування 75
3.8.2 Контейнеризація за допомогою Docker 76
3.9 Висновки до розділу 3 77
ВИСНОВКИ 79
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 81

ВСТУП

Актуальність теми

Сучасне машинобудування переживає етап активної цифровізації та автоматизації виробничих процесів. Верстати з числовим програмним керуванням (ЧПК) стали невід'ємною частиною будь-якого машинобудівного підприємства, забезпечуючи високу точність обробки, повторюваність результатів та можливість виготовлення деталей складної геометрії. За даними аналітичних досліджень, світовий ринок верстатів з ЧПК демонструє стабільне зростання, що свідчить про постійне збільшення попиту на автоматизоване металообробне обладнання.

Ефективність роботи на верстатах з ЧПК значною мірою залежить від правильного вибору режимів різання. Базові принципи теорії різання металів та проєктування різальних інструментів фундаментально розкриті в працях Г. І. Грановського [1], П. Р. Родіна [2], Ю. М. Внукова [3], І. М. Пижова та Г. П. Клименка [4]. Режим різання — це сукупність параметрів, що визначають умови обробки матеріалу: частота обертання шпинделя, подача інструменту та глибина різання за один прохід. Неправильний вибір цих параметрів може призвести до передчасного зносу дорогого різального інструменту, погіршення якості поверхні, збільшення часу виготовлення деталі та пошкодження верстата.

Традиційно оператори верстатів з ЧПК користуються табличними довідниками для вибору режимів різання. Проте використання паперових довідників має суттєві недоліки: вони громіздкі, інформація в них швидко застаріває, а робота з ними безпосередньо біля верстата є незручною. Існуючі програмні рішення для підбору режимів різання, такі як CAM-системи, архітектура та принципи роботи яких детально висвітлені В. М. Третьяковим [5], мають високу вартість ліцензій та вимагають спеціалізованих навичок.

Месенджер Telegram став однією з найпопулярніших платформ для комунікації. Розробка Telegram-бота для пошуку режимів різання є

актуальною задачею. Практичні аспекти розробки чат-ботів для автоматизації бізнес-процесів розглядаються А. О. Павловим [6]. Застосування інформаційних технологій у виробничому середовищі загалом спирається на принципи побудови інформаційних систем у менеджменті, висвітлені В. Ф. Сітніком [7]. Інформаційна система у вигляді Telegram-бота здатна забезпечити операторам швидкий доступ до бази даних безпосередньо з мобільного пристрою, а також функцію обміну досвідом.

Таким чином, розробка інформаційної системи у вигляді Telegram-бота для пошуку режимів різання на верстатах з ЧПК є актуальною науково-практичною задачею, що має значний потенціал для впровадження у виробничу практику.

Мета дослідження

Метою кваліфікаційної роботи є розробка інформаційної системи у вигляді Telegram-бота для пошуку, управління та обміну режимами різання на верстатах з числовим програмним керуванням. Система має підтримувати функції пошуку, фільтрації, оцінювання та коментування режимів різання, а також надавати можливість користувачам пропонувати власні режими обробки з подальшою модерацією.

Завдання дослідження

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- Проаналізувати предметну область, дослідити особливості вибору режимів різання на верстатах з ЧПК.
- Провести огляд існуючих програмних рішень для підбору режимів різання, обґрунтувати доцільність розробки нового продукту.
- Дослідити сучасні технології розробки Telegram-ботів та серверних застосунків.

- Спроєктувати архітектуру інформаційної системи з урахуванням об'єктно-орієнтованих патернів проєктування та шаблонів корпоративних застосунків, описаних Е. Гаммою [8] та М. Фаулером [9].
- Розробити структуру бази даних для зберігання інформації про матеріали, інструменти та режими різання.
- Реалізувати серверну частину застосунку з використанням офіційної документації Telegram Bot API [10], бібліотеки grammY [11] та фреймворку NestJS [12]. Для роботи з даними інтегрувати Prisma ORM [13] та СУБД PostgreSQL [14] на базі мови TypeScript [15].
- Розробити систему модерації користувацького контенту.
- Провести тестування розробленого програмного забезпечення.

Об’єкт дослідження

Процес інформаційної підтримки операторів верстатів з числовим програмним керуванням при виборі оптимальних режимів різання для обробки різних матеріалів.

Предмет дослідження

Предметом дослідження є методи та засоби розробки інформаційних систем на базі платформи Telegram для автоматизації пошуку та управління технологічною інформацією у сфері металообробки.

Методи дослідження

У процесі виконання кваліфікаційної роботи застосовано комплекс наукових методів дослідження, поєднання яких забезпечило системність розробки програмного продукту. Теоретичний базис дослідження сформовано на основі методу аналізу та декомпозиції предметної області, синтезу та класифікації для структуризації масивів даних щодо конструкційних матеріалів та режимів їх обробки, а також порівняльного аналізу та абстрагування при виборі оптимального технологічного стеку і побудові

концептуальної архітектури. Практична реалізація та верифікація системи базувалися на методах об'єктно-орієнтованого проєктування для формування структури програмного коду, інструментах UML-моделювання при створенні схем баз даних і сценаріїв взаємодії компонентів, а також на емпіричних методах тестування для верифікації та налагодження розробленого програмного забезпечення.

Практичне значення одержаних результатів

Практичне значення одержаних результатів дослідження полягає у створенні готового до впровадження програмного продукту, який може бути використаний операторами верстатів з ЧПК для підвищення ефективності їх роботи.

Розроблена інформаційна система дозволяє:

- швидко знаходити рекомендовані режими різання для конкретної комбінації матеріалу та інструменту;
- оцінювати ефективність режимів на основі досвіду інших користувачів через систему рейтингів;
- обмінюватися практичним досвідом через систему коментарів;
- зберігати найбільш корисні режими у списку обраного для швидкого доступу;
- пропонувати власні напрацювання для поповнення бази даних.

Результати дослідження можуть бути використані на машинобудівних підприємствах різного масштабу для підвищення продуктивності праці операторів верстатів з ЧПК, зменшення кількості браку та оптимізації витрат на різальний інструмент.

Крім того, розроблена архітектура системи та використані технологічні рішення можуть слугувати основою для створення аналогічних інформаційних систем в інших галузях, де існує потреба у швидкому доступі до структурованої технічної інформації.

Апробація результатів

Основні положення та результати дослідження були представлені на Міжнародній інтернет-конференції здобувачів вищої освіти і молодих учених «Інформаційні технології: теорія і практика» (Харків-Запоріжжя-Дніпро, 25–27 березня 2026р.) [17] та конференції студентів, аспірантів і молодих вчених ЗНУ «Молода наука» (Запоріжжя, 2026 р.) [16] а також обговорювалися на студентських наукових семінарах кафедри електроніки, інформаційних систем та програмного забезпечення.

Глосарій

Верстат з ЧПК (CNC — Computer Numerical Control) — металорізальний верстат, обладнаний системою числового програмного керування, яка забезпечує автоматичне виконання заданої послідовності технологічних операцій відповідно до керуючої програми.

Режим різання — сукупність параметрів процесу механічної обробки матеріалу, що включає частоту обертання шпинделя, подачу інструменту та глибину різання за один прохід.

Частота обертання шпинделя (n) — швидкість обертання шпинделя верстата, що вимірюється в обертах за хвилину (об/хв, RPM). Визначає швидкість різання та впливає на якість обробленої поверхні.

Подача (f) — швидкість переміщення різального інструменту відносно заготовки, що вимірюється в міліметрах за хвилину (мм/хв) або міліметрах на оберт (мм/об).

Глибина різання (t) — товщина шару матеріалу, що знімається за один прохід інструменту, вимірюється в міліметрах (мм).

Різальний інструмент — знаряддя для механічної обробки матеріалів різанням. Включає фрези, свердла, різці, мітчики та інші типи інструментів.

Фреза — багатолезовий різальний інструмент для обробки плоских та фасонних поверхонь, пазів, уступів тощо.

Telegram-бот — програмний застосунок, що функціонує в межах месенджера Telegram та забезпечує автоматизовану взаємодію з користувачами через текстові команди, кнопки та інші елементи інтерфейсу.

API (Application Programming Interface) — програмний інтерфейс застосунку, що визначає способи взаємодії між різними програмними компонентами або системами.

Telegram Bot API — офіційний програмний інтерфейс платформи Telegram для створення та управління ботами.

Backend (серверна частина) — компонент програмної системи, що відповідає за обробку бізнес-логіки, роботу з базою даних та взаємодію з зовнішніми сервісами.

База даних — організована сукупність структурованих даних, що зберігається в електронному вигляді та забезпечує ефективний доступ до інформації.

ORM (Object-Relational Mapping) — технологія програмування, що забезпечує зв'язок між об'єктною моделлю програми та реляційною базою даних.

Модерація контенту — процес перевірки та контролю користувацького контенту перед його публікацією в системі з метою забезпечення якості та відповідності встановленим правилам.

TypeScript — мова програмування, що є надмножиною JavaScript з підтримкою статичної типізації, яка компілюється в JavaScript.

NestJS — прогресивний фреймворк для створення ефективних та масштабованих серверних застосунків на платформі Node.js.

grammY — сучасна бібліотека для розробки Telegram-ботів на TypeScript/JavaScript, що надає зручний API для роботи з Telegram Bot API.

Prisma — сучасний ORM для Node.js та TypeScript, що забезпечує типобезпечний доступ до бази даних.

PostgreSQL — потужна об'єктно-реляційна система управління базами даних з відкритим вихідним кодом.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Характеристика процесу обробки на верстатах з ЧПК

Верстати з числовим програмним керуванням являють собою складні технічні системи, що поєднують механічні вузли, електронні компоненти та програмне забезпечення для автоматизованого виконання технологічних операцій обробки матеріалів. Історія розвитку верстатів з ЧПК розпочалася у середині XX століття, коли в Массачусетському технологічному інституті було створено перший прототип верстата з числовим керуванням. З того часу технологія пройшла значний шлях розвитку, і сьогодні верстати з ЧПК є основою сучасного машинобудівного виробництва.

Принцип роботи верстата з ЧПК базується на виконанні послідовності команд, записаних у керуючій програмі. Ця програма, зазвичай написана мовою G-кодів, містить інформацію про траєкторію руху інструменту, швидкості переміщення, режими обробки та допоміжні функції. Система керування верстата інтерпретує ці команди та забезпечує їх точне виконання через приводи осей та шпинделя.

Сучасні верстати з ЧПК поділяються на декілька основних типів залежно від виду виконуваних операцій.

Фрезерні верстати з ЧПК

Призначені для обробки плоских та фасонних поверхонь за допомогою обертового інструменту — фрези. Вони широко застосовуються для виготовлення корпусних деталей, прес-форм, штампів та інших виробів складної геометрії. Фрезерні верстати можуть мати від трьох до п'яти керованих осей, що дозволяє обробляти деталі з різних сторін без переустановлення.

Токарні верстати з ЧПК

Використовуються для обробки тіл обертання. На відміну від фрезерних верстатів, тут обертається заготовка, а інструмент виконує лінійні

переміщення. Токарні центри з ЧПК можуть комплектуватися приводним інструментом, що дозволяє виконувати фрезерні та свердлильні операції без переустановлення деталі.

Свердлильні верстати з ЧПК

Спеціалізуються на виконанні отворів різного діаметра та глибини. Вони часто використовуються у серійному виробництві для обробки деталей з великою кількістю отворів.

Багатоцільові обробні центри

Поєднують можливості декількох типів верстатів, дозволяючи виконувати фрезерні, свердлильні, розточувальні та інші операції на одному обладнанні. Це суттєво скорочує час виготовлення деталей та підвищує точність обробки за рахунок зменшення кількості переустановлень.

Ефективність роботи на верстатах з ЧПК визначається багатьма факторами, серед яких правильний вибір режимів різання займає одне з ключових місць. Режим різання — це комплексний параметр, що характеризує умови процесу механічної обробки та включає три основні складові.

Швидкість різання V

Це швидкість переміщення точки різальної кромки інструменту відносно оброблюваної поверхні. Вона вимірюється в метрах за хвилину (м/хв) та розраховується за формулою: $V = \frac{\pi * D * n}{1000}$ де D — діаметр інструменту або заготовки в міліметрах, n — частота обертання шпинделя в об/хв. Швидкість різання суттєво впливає на температуру в зоні обробки, стійкість інструменту та якість обробленої поверхні.

Подача f

Характеризує швидкість переміщення інструменту відносно заготовки. Для фрезерної обробки часто використовують хвилинну подачу (мм/хв), яка визначає, на яку відстань переміститься інструмент за одну хвилину. Також застосовують подачу на зуб (мм/зуб) — величину переміщення інструменту за час, поки один зуб фрези перебуває в контакті з матеріалом. Подача впливає

на продуктивність обробки, навантаження на інструмент та шорсткість обробленої поверхні.

Глибина різання t

Це товщина шару матеріалу, що знімається за один прохід інструменту. Вона вимірюється в міліметрах та визначає припуск, що видаляється за одну операцію. Глибина різання впливає на сили різання, вібрації в системі та ймовірність поломки інструменту.

Вибір оптимальних режимів різання є складною інженерною задачею, що залежить від численних факторів:

Властивості оброблюваного матеріалу

Мають визначальний вплив на вибір режимів. Твердість, міцність, теплопровідність та схильність до наклепу матеріалу визначають допустимі швидкості різання та навантаження на інструмент. Наприклад, при обробці загартованих сталей швидкість різання суттєво знижується порівняно з обробкою алюмінієвих сплавів.

Характеристики різального інструменту

Включають геометрію ріжучої частини, матеріал інструменту, наявність та тип покриття, кількість зубів (для фрез). Твердосплавний інструмент дозволяє працювати на значно вищих швидкостях, ніж інструмент зі швидкорізальної сталі. Сучасні покриття (TiN, TiAlN, AlCrN та інші) підвищують стійкість інструменту та дозволяють збільшити режими обробки.

Жорсткість технологічної системи

Жорсткість технологічної системи *ВПІД* (верстат — пристосування — інструмент — деталь) обмежує максимально допустимі режими різання. При недостатній жорсткості виникають вібрації, що призводять до погіршення якості обробки та прискореного зносу інструменту.

Вимоги до якості обробленої поверхні

Визначають межі подачі та глибини різання. Для отримання низької шорсткості зазвичай застосовують знижені подачі та малі глибини різання при підвищених швидкостях.

Потужність приводу верстата

Обмежує максимальний об'єм матеріалу, що може бути видалений за одиницю часу. При перевищенні допустимих навантажень можливе спрацювання захисту або пошкодження обладнання.

Традиційний підхід до визначення технологічних параметрів механічної обробки базується на використанні паперових табличних довідників, зокрема загальномашинобудівних нормативів, каталогів виробників інструментального оснащення та профільних галузевих стандартів, які містять емпірично обґрунтовані рекомендації для типових комбінацій конструкційних матеріалів і різальних інструментів. Проте експлуатація друкованих носіїв у сучасних високотехнологічних виробництвах супроводжується низкою критичних обмежень, серед яких найвагомішими є тривалий час пошуку взаємопов'язаних параметрів, швидке моральне застарівання інформаційного фонду в умовах появи нових сплавів і геометрій інструменту, а також узагальнений характер нормативів, що унеможливлює точне врахування специфіки та динамічних характеристик конкретного верстатного обладнання. Крім того, статичний формат паперових видань виключає можливість інтеграції та оперативного обміну практичним досвідом між операторами, а безпосереднє використання масивних друкованих томів у специфічних цехових умовах біля пультів керування верстатів з ЧПК є деструктивним з погляду ергономіки. Зазначені недоліки обумовлюють об'єктивну необхідність проєктування та впровадження сучасних мобільних інформаційних систем, орієнтованих на ефективну цифрову підтримку операторів при виборі та оптимізації режимів різання безпосередньо на робочому місці.

1.2 Огляд літературних джерел

В ході аналізу предметної області проведено огляд існуючих літературних джерел, що відносяться до теми кваліфікаційної роботи. У

процесі огляду було визначено основні підходи, методи та технології, що застосовуються у сфері автоматизації вибору режимів різання, розробки інформаційних систем та створення Telegram-ботів.

Фундаментальні питання теорії різання металів розглянуто у класичних роботах вітчизняних та зарубіжних вчених. У підручнику «Різання металів» [1] авторства Грановського Г.І. та Грановського В.Г. викладено теоретичні основи процесу різання, розглянуто фізичні явища, що супроводжують обробку матеріалів, та наведено методики розрахунку режимів різання. Автори детально аналізують вплив різних факторів на процес обробки, включаючи властивості оброблюваного матеріалу, геометрію інструменту та умови охолодження. Ця робота є базовою для розуміння принципів формування рекомендацій щодо режимів обробки.

Сучасні підходи до проєктування різального інструменту представлено у роботі Родіна П.Р. «Основи проєктування різальних інструментів» [2]. Автор розглядає принципи конструювання фрез, свердел, різців та інших типів інструментів з урахуванням умов їх експлуатації. Особливу увагу приділено взаємозв'язку геометричних параметрів інструменту та рекомендованих режимів обробки. Ці знання є важливими для формування структури бази даних інструментів у розроблюваній системі.

Питання оптимізації режимів різання з урахуванням сучасних вимог до продуктивності та якості обробки досліджено у роботі Внукова Ю.М. «Теорія різання» [3]. Автор пропонує комплексний підхід до вибору параметрів обробки, що враховує не лише технологічні, але й економічні фактори. Розглянуто методи визначення оптимальних режимів за критеріями мінімальної собівартості, максимальної продуктивності та заданої якості обробки.

У навчальному посібнику Пижова І.М. та Клименка Г.П. «Теорія різання матеріалів та різальний інструмент» [4] систематизовано сучасні знання про процеси механічної обробки. Автори приділяють значну увагу особливостям обробки різних груп матеріалів: сталей, чавунів, алюмінієвих та титанових

сплавів, композиційних матеріалів. Ця інформація є корисною для формування каталогу матеріалів у розроблюваній системі.

Питання застосування комп'ютерних технологій у машинобудуванні розглянуто у монографії Третьякова В.М. «CAM-системи в машинобудуванні» [5]. Автор аналізує можливості сучасних CAM-систем для автоматизації підготовки керуючих програм, включаючи автоматичний вибір режимів різання на основі вбудованих баз даних. Розглянуто принципи побудови технологічних баз даних та алгоритми вибору параметрів обробки.

Офіційна документація Telegram Bot API [6] є основним джерелом інформації для розробки Telegram-ботів. Документація містить повний опис можливостей платформи, методів взаємодії з користувачами, типів повідомлень та елементів інтерфейсу. Особливу увагу приділено механізмам inline-клавіатур та callback-запитів, які забезпечують інтерактивну взаємодію користувача з ботом.

Практичні аспекти розробки чат-ботів для бізнесу розглянуто у посібнику Павлова А.О. «Розробка чат-ботів для бізнесу» [7]. Автор аналізує різні сценарії використання ботів, принципи проєктування діалогового інтерфейсу та методи підвищення залученості користувачів. Наведено рекомендації щодо структурування інформації та організації навігації у боті.

Бібліотека grammY для розробки Telegram-ботів на TypeScript детально документована в офіційній документації [8]. GrammY надає сучасний, типобезпечний API для роботи з Telegram Bot API, підтримує middleware-архітектуру та забезпечує зручну обробку різних типів оновлень. Бібліотека активно розвивається та має велику спільноту розробників.

Фреймворк NestJS [9] є прогресивним інструментом для створення серверних застосунків на платформі Node.js. Офіційна документація надає вичерпну інформацію про архітектуру фреймворку, принципи модульності, dependency injection, декоратори та інші концепції. NestJS підтримує TypeScript за замовчуванням та забезпечує масштабовану структуру проєкту.

ORM Prisma [10] є сучасним інструментом для роботи з базами даних у Node.js та TypeScript проєктах. Документація описує принципи роботи з Prisma Schema для визначення моделей даних, Prisma Client для типобезпечного доступу до бази даних та Prisma Migrate для управління міграціями. Prisma забезпечує автоматичну генерацію TypeScript типів на основі схеми бази даних.

Архітектурні патерни для побудови корпоративних застосунків систематизовано у класичній роботі Фаулера М. «Шаблони корпоративних застосунків» [11]. Автор описує різні підходи до організації бізнес-логіки, роботи з даними та побудови рівня представлення. Ці патерни є корисними для проєктування архітектури розроблюваної системи.

Офіційна документація PostgreSQL [12] є основним джерелом інформації про можливості цієї системи управління базами даних. PostgreSQL є потужною об'єктно-реляційною СУБД з відкритим вихідним кодом, що підтримує складні запити, транзакції, індекси та інші функції, необхідні для ефективної роботи з даними.

У підручнику Сітніка В.Ф. «Інформаційні системи в менеджменті» [13] розглянуто загальні принципи побудови інформаційних систем, їх класифікацію та методології розробки. Автор аналізує етапи життєвого циклу інформаційних систем та критерії оцінки їх ефективності.

Офіційна документація TypeScript [14] описує можливості мови програмування, що використовується у проєкті. TypeScript є надмножиною JavaScript з підтримкою статичної типізації, що дозволяє виявляти помилки на етапі компіляції та забезпечує кращу підтримку з боку IDE.

Класичні патерни проєктування описано у книзі «Патерни проєктування» [15] авторів Гамма Е., Хелм Р. та інших. Автори систематизували перевірені рішення типових проблем проєктування програмного забезпечення. Знання цих патернів є важливим для створення якісної та підтримуваної архітектури.

Аналіз літературних джерел показує, що питання вибору режимів різання є актуальною та добре дослідженою проблемою в галузі машинобудування. Проте існуючі рішення для автоматизації цього процесу мають суттєві обмеження: CAM-системи вимагають значних фінансових витрат та кваліфікації користувача, онлайн-калькулятори виробників інструменту обмежені власним асортиментом, а табличні довідники є незручними у використанні.

Сучасні технології розробки веб-застосунків та Telegram-ботів надають широкі можливості для створення зручних та доступних інформаційних систем. Поєднання цих технологій з накопиченими знаннями у сфері металообробки дозволяє створити інноваційний продукт, що відповідатиме потребам операторів верстатів з ЧПК.

1.3 Аналіз програмних продуктів-аналогів

Проблематика вибору режимів різання є актуальною для машинобудівної галузі, тому на ринку існує ряд програмних рішень різного рівня складності та вартості. Для обґрунтування доцільності розробки нового програмного продукту проведено аналіз існуючих рішень, визначено їх переваги та недоліки.

1.3.1 CAM-системи провідних розробників

Autodesk Fusion 360 є хмарною платформою для проєктування та виробництва, що включає потужний CAM модуль. Система автоматично розраховує режими різання на основі вбудованої бази даних матеріалів та інструментів. Користувач може обрати матеріал заготовки та інструмент із каталогу, а система запропонує рекомендовані параметри обробки.

Проведений аналіз функціональних можливостей програмного комплексу Autodesk Fusion 360 дозволив визначити його ключові переваги та недоліки в контексті вирішення інженерних завдань. Головною перевагою

даної платформи є забезпечення комплексного підходу до наскрізного проєктування та виробництва, що інтегрує CAD/CAM/CAE-середовища в межах єдиного цифрового простору. Завдяки хмарній архітектурі системи реалізовано оперативний доступ до проєктних даних з різних апаратних пристроїв, а також забезпечено автоматичне й регулярне оновлення інтегрованих баз даних конструкційних матеріалів та різальних інструментів. Додатковим стимулюючим фактором для впровадження системи в освітній процес та діяльність початкових підприємств є наявність безкоштовної ліцензії для навчальних закладів і стартапів.

Водночас використання Autodesk Fusion 360 супроводжується низкою суттєвих обмежень, серед яких першочергове значення має висока вартість комерційного ліцензування, що становить від 500 USD на рік за одне робоче місце. Складність інтерфейсу та широкий інструментарій вимагають значних часових ресурсів на освоєння програмного продукту інженерно-технічним персоналом. Крім того, специфіка хмарної технології зумовлює жорстку залежність від постійного та стабільного підключення до мережі Інтернет. З погляду оперативного підбору технологічних параметрів обробки, архітектура комплексу володіє значною надлишковою функціональністю. Окремим критичним недоліком у межах даного дослідження є повна відсутність орієнтації інтерфейсу на мобільне використання безпосередньо в цехових умовах біля технологічного обладнання з ЧПК.

Програмний комплекс Mastercam позиціонується як одне з найбільш поширених та перевірених часом світових рішень у сегменті CAM-систем, володіючи понад сорокарічною історією розвитку та експлуатації. Платформа містить об'ємну інтегровану базу даних режимів різання, яка охоплює широкий спектр комбінацій конструкційних матеріалів та різальних інструментів. Функціонал системи дозволяє операторам та технологам систематизувати власні виробничі напрацювання, а також формувати персоналізовані бібліотеки інструментального оснащення.

До фундаментальних переваг Mastercam належать висока технологічна потужність та ефективність при проєктуванні складних видів механічної обробки, зокрема багатоосьового фрезерування та точіння. Наявність масштабної вбудованої довідкової бази супроводжується гнучкими інструментами кастомізації інженерного середовища під специфічні потреби конкретного промислового підприємства. Крім того, тривала присутність на ринку зумовила формування численного експертного співтовариства та розвиненої інфраструктури технічної підтримки, що значно спрощує вирішення складних технологічних завдань.

Попри очевидні переваги, впровадження та експлуатація Mastercam пов'язані з низкою вагомих обмежень економічного та організаційного характеру. Основним деструктивним фактором є критично висока вартість комерційного ліцензування, яка становить кілька тисяч доларів США за одне робоче місце, що обмежує доступність продукту для малого бізнесу чи індивідуальних операторів. Специфічний, перевантажений елементами керування інтерфейс зумовлює високий поріг входження та вимагає тривалого цільового навчання інженерного персоналу. З технічного погляду, архітектура системи розрахована виключно на функціонування в межах стаціонарних настільних обчислювальних станцій і має надлишкову функціональність, якщо кінцевою метою користувача є швидкий локальний підбір технологічних режимів безпосередньо біля обладнання з ЧПК.

Програмний комплекс SolidCAM є спеціалізованим CAM-рішенням, що функціонує на основі глибокої безшовної інтеграції в інженерне середовище SolidWorks. Ключовою особливістю даної платформи є застосування запатентованої інноваційної технології iMachining, яка в автоматичному режимі здійснює генерацію оптимальних траєкторій руху інструменту та розрахунок відповідних режимів різання, динамічно враховуючи поточне термічне та механічне навантаження на різальну кромку.

До вагомих переваг SolidCAM належить повна асоціативність із геометричними моделями CAD-платформи SolidWorks, завдяки чому будь-які

зміни в конструкції деталі автоматично відображаються на технологічних переходах обробки без потреби повторного проєктування. Вищезгадана технологія оптимізації дозволяє суттєво скоротити цикл виробництва та подовжити термін експлуатації інструментального оснащення за рахунок автоматизованого інтелектуального аналізу фізичних процесів різання на кожній ділянці траєкторії.

Основним деструктивним фактором при впровадженні SolidCAM є його жорстка архітектурна залежність від екосистеми SolidWorks, що унеможливлює автономне використання продукту або його експлуатацію в поєднанні з іншими поширеними CAD-системами без додаткового конвертування файлів. Крім того, комплекс характеризується високою вартістю комерційних ліцензій та підвищеними вимогами до кваліфікації інженерного персоналу, що зумовлює складність і тривалість процесу первинного освоєння системи та робить її економічно неефективною для розв'язання базових цехових завдань з оперативного підбору режимів обробки на робочому місці оператора ЧПК.

1.3.2 Онлайн-калькулятори виробників інструменту

Онлайн-сервіс CoroPlus ToolGuide, розроблений провідним світовим виробником металорізального інструменту Sandvik Coromant, є спеціалізованою вебплатформою для автоматизованого підбору інструментального оснащення та розрахунку технологічних параметрів механічної обробки. Функціональні алгоритми системи здійснюють багатокритеріальний аналіз вхідних даних, інтегруючи у розрахункову модель тип запланованої технологічної операції, фізико-механічні властивості матеріалу заготовки, кінематичну потужність верстатного обладнання, а також геометричні параметри деталі.

Головною перевагою даного інструментального рішення є надання безкоштовного доступу до обчислювальних ресурсів платформи для широкого кола інженерно-технічних працівників. Сервіс генерує високоточні та

технологічно обґрунтовані рекомендації щодо режимів різання, які безпосередньо спираються на масштабну еталонну базу даних та результати натурних випробувань розробника. Сучасний адаптивний веб-інтерфейс забезпечує високу ергономічність взаємодії користувача з платформою, а хмарна архітектура гарантує регулярне й автоматичне оновлення номенклатури інструментів та алгоритмів оптимізації без потреби локального втручання в програмне забезпечення.

Основним деструктивним фактором при експлуатації CoroPlus ToolGuide є сувора комерційна обмеженість системи виключно асортиментом продукції Sandvik Coromant. Застосунок повністю ігнорує номенклатуру різального інструменту інших світових виробників, що значно звужує сферу його універсального застосування на багатопрофільних машинобудівних підприємствах. З погляду краудсорсингу та накопичення знань, архітектура сервісу є закритою — у ній відсутня будь-яка можливість соціальної взаємодії чи обміну практичним досвідом між операторами. Крім того, відсутність спеціалізованого мобільного застосунку, оптимізованого для специфічних умов цеху, ускладнює оперативне використання платформи безпосередньо біля технологічних комплексів з ЧПК.

Спеціалізований інформаційний ресурс Kennametal NOVO є аналогічним цифровим рішенням, розробленим компанією Kennametal для автоматизації процесів вибору інструментального оснащення та обчислення оптимальних параметрів механічної обробки. Функціональне ядро платформи дозволяє виконувати розширене інженерне моделювання, що включає не лише базовий підбір режимів, а й комплексний розрахунок кінематичної потужності верстата та сил, що діють у зоні різання.

Головною перевагою даного сервісу є висока точність та деталізація інженерно-технічних обчислень, що дозволяє мінімізувати ризики поломки обладнання на етапі підготовки виробництва. Крім того, інтерфейс системи містить аналітичний інструментарій для компаративного аналізу (порівняння) декількох альтернативних варіантів обробки за критеріями продуктивності та

економічної ефективності. Важливим стимулюючим фактором для кінцевих користувачів є надання повністю безкоштовного доступу до всіх розрахункових модулів платформи.

Водночас експлуатація Kennametal NOVO супроводжується низкою обмежень, зумовлених її маркетинговою орієнтацією та архітектурною побудовою. Сервіс суворо лімітований виключно власною номенклатурою продукції компанії Kennametal, що унеможливлює інтеграцію інструментів від інших виробників у єдиний розрахунковий цикл. Специфічна логіка побудови інтерфейсу, орієнтована на глибоке проєктування, є надмірно складною для оперативного вирішення базових утилітарних завдань безпосередньо в цеху. Окремим недоліком є відсутність оптимізованої мобільної версії застосунку, що обмежує автономність використання платформи безпосередньо біля пультів керування верстатів з ЧПК.

Walter GPS (Global Product Selector) система підбору інструменту та режимів від компанії Walter. Сервіс орієнтований на промислових користувачів та надає детальні рекомендації для складних операцій.

Подібні обмеження мають калькулятори інших виробників: Seco Tools, Iscar, Mitsubishi Materials, Kyocera тощо. Усі вони орієнтовані на просування власної продукції та не надають універсальних рекомендацій.

1.3.3 Універсальні онлайн-калькулятори

В Інтернеті існує велика кількість безкоштовних калькуляторів режимів різання різного рівня якості. Більшість з них базуються на спрощених формулах та табличних даних.

Інформаційний ресурс CNC Cookbook позиціонується як поширене довідково-освітнє середовище для операторів верстатів з ЧПК, що об'єднує в межах єдиної платформи калькулятори технологічних режимів, інтегровані посібники з програмування на базі G-коду та розгалужену базу навчальних матеріалів. Функціональне ядро калькулятора дозволяє здійснювати розрахунок базових параметрів обробки на основі лінійного аналізу введених

користувачем даних. До очевидних переваг ресурсу належать надання безкоштовного доступу до обчислювальних модулів, висока ергономічність і простота користувацького інтерфейсу, а також наявність потужного супутнього контенту для підвищення кваліфікації інженерного персоналу.

Водночас експлуатація CNC Cookbook супроводжується низкою обмежень, серед яких звужений обсяг довідкової бази даних щодо специфічних конструкційних матеріалів, виключно англомовна локалізація інтерфейсу, що створює мовний бар'єр для частини вітчизняних фахівців, та повна відсутність інструментарію соціальної взаємодії чи обміну практичним досвідом. Окремим деструктивним фактором, який знижує зручність використання платформи, є висока концентрація комерційного рекламного контенту.

Мобільний застосунок FSWizard є спеціалізованим програмним рішенням, орієнтованим на оперативне обчислення параметрів фрезерної та токарної обробки безпосередньо в умовах виробничих цехів. Кросплатформна сумісність із мобільними операційними системами iOS та Android у поєднанні з архітектурною можливістю автономного функціонування (офлайн-режим) без підключення до мережі Інтернет забезпечує високу мобільність та автономність інженера біля технологічного обладнання. Крім того, застосунок володіє інтуїтивно зрозумілим лаконічним інтерфейсом, що мінімізує час на первинне освоєння продукту.

Проте функціональний потенціал FSWizard суттєво обмежений комерційною політикою розробника: безкоштовна версія продукту має штучно урізаний інструментарій, а архітектура додатка не містить інтегрованих баз даних конкретних світових виробників інструментального оснащення, що змушує користувача вводити геометричні параметри вручну. Як і в попередньому випадку, дане рішення є суто утилітарним калькулятором і повністю ізольоване від концепції краудсорсингу, оскільки в ньому не передбачено механізмів комунікації та обміну практичними напрацюваннями між членами професійної спільноти.

1.3.4 Табличні довідники та довідкова література

Традиційні друковані табличні довідники, серед яких найбільш поширеними у вітчизняній інженерній практиці є багатотомні видання класичного типу (зокрема, фундаментальні праці з технології машинобудування та нормативні збірники режимів різання металів), залишаються стабільним елементом інформаційного забезпечення на багатьох промислових підприємствах. Дані джерела містять глибоко систематизовану й емпірично обґрунтовану інформацію про регламентовані технологічні параметри обробки для базових, стандартизованих комбінацій конструкційних матеріалів та різальних інструментів.

До беззаперечних переваг традиційних довідників належить високий рівень структурованості та системності викладу матеріалу, який спирається на єдину методологічну базу. Крім того, паперові носії мають абсолютну автономність, оскільки не потребують наявності обчислювальної техніки, стабільного електроживлення чи підключення до мережевих ресурсів, а наведені в них нормативні дані є верифікованими й перевіреними тривалим часом практичної експлуатації в реальному секторі виробництва.

Поряд із цим, використання класичних паперових аналогів у сучасних високотехнологічних умовах супроводжується низкою критичних обмежень. Ключовим недоліком є низька оперативність та значна незручність процедури пошуку інформації, яка вимагає від інженера ручного перегляду багатьох взаємопов'язаних таблиць, розгалужених графіків та коефіцієнтів. Фізична громіздкість паперових томів суттєво ускладнює їх безпосередню експлуатацію в умовах виробничих цехів біля верстатного обладнання. З інженерного погляду, найвагомішим мінусом є прогресуюча застарілість інформаційної бази: статичний формат друкованого видання унеможливлює оперативне врахування сучасних інноваційних сплавів, композитів та прогресивних геометричних конфігурацій різального інструменту, що з'являються на ринку. Наостанок, архітектура друкованого довідника є статичною та замкненою, що повністю виключає можливість динамічної

інтеграції реального практичного досвіду операторів або оперативного обміну напрацьованими технологічними рішеннями в межах професійної спільноти.

1.3.5 Telegram-боти та месенджер-рішення

Аналіз сучасного стану ринку програмного забезпечення та засобів автоматизації інженерних обчислень дозволив констатувати повну відсутність на момент проведення дослідження спеціалізованих Telegram-ботів, призначених для підбору режимів різання з україномовною локалізацією інтерфейсу. Наявні на сьогодні функціональні рішення в межах сучасних месенджерів орієнтовані переважно на інші сектори, такі як електронна комерція, сфера послуг та загальна освіта, або ж характеризуються суттєво обмеженим інструментарієм і фрагментарною інформаційною базою.

Виявлений інформаційно-технологічний вакуум створює об'єктивні передумови для розробки та впровадження інноваційного програмного продукту, архітектура якого дозволить інтегрувати високий рівень ергономічності й оперативність мобільного доступу до обчислювальних модулів безпосередньо через інтерфейс поширеного месенджера зі структурованою й масштабованою реляційною базою даних, що містить регламентовані режими різання для диференційованих груп конструкційних матеріалів та інструментального оснащення. Крім того, перспективна система має поєднувати в собі інтерактивний інструментарій оцінювання, рецензування та коментування запропонованих режимів на основі реального емпіричного досвіду виробничого персоналу, механізм децентралізованого поповнення інформаційного фонду за рахунок легітимного користувацького контенту, а також обов'язкову багаторівневу адміністративну модерацію вхідних даних для забезпечення високої якості, достовірності та верифікації технологічної інформації.

1.3.6 Порівняльна таблиця рішень

На основі проведеного аналізу складено порівняльну таблицю існуючих рішень (табл. 1.1).

Таблиця *1.1 — Порівняльний аналіз існуючих рішень*

Критерій	**CAM-системи**	**Калькулятори виробників**	**Універсальні калькулятори**	**Довідники**
Вартість	Висока	Безкоштовно	Безкоштовно / низька	Низька
Мобільний доступ	Ні	Частково	Частково	Ні
Універсальність	Висока	Низька	Середня	Середня
Обмін досвідом	Ні	Ні	Ні	Ні
Простота використання	Низька	Середня	Висока	Низька
Актуальність даних	Висока	Висока	Середня	Низька
Українська мова	Ні	Ні	Ні	Частково

Аналіз показує, що жодне з існуючих рішень не забезпечує повного набору необхідних функцій: мобільної доступності, простоти використання, можливості обміну практичним досвідом та підтримки української мови. Це обґрунтовує доцільність розробки нового програмного продукту.

1.4 Постановка завдання

На основі проведеного аналізу предметної області, огляду літературних джерел та дослідження існуючих програмних рішень сформульовано вимоги до розроблюваної інформаційної системи.

1.4.1 Мета розробки

Метою розробки є створення інформаційної системи у вигляді Telegram-бота, що забезпечить операторам верстатів з ЧПК зручний та швидкий доступ до бази даних рекомендованих режимів різання. Система повинна підтримувати пошук режимів за комбінацією матеріалу та інструменту, оцінювання та коментування режимів, а також можливість додавання користувацького контенту з модерацією.

1.4.2 Функціональні вимоги

Система повинна забезпечувати наступну функціональність:

Для користувачів:

- Пошук режимів різання. Користувач має можливість знайти рекомендовані режими обробки, обравши матеріал заготовки та тип інструменту. Результати пошуку повинні включати значення частоти обертання шпинделя, подачі та глибини різання за прохід.
- Перегляд каталогу матеріалів. Система надає доступ до структурованого каталогу матеріалів, організованого за категоріями (сталі, алюмінієві сплави, пластики тощо). Для кожного матеріалу доступна інформація про його характеристики.
- Перегляд каталогу інструментів. Користувач може переглядати каталог різальних інструментів з детальною інформацією про кожен: тип, діаметр, довжина, кількість зубів (для фрез), матеріал та покриття.

- Оцінювання режимів. Користувачі можуть оцінювати режими різання за допомогою системи лайків та дизлайків. Це дозволяє визначити найбільш ефективні режими на основі колективного досвіду.
- Коментування режимів. Система надає можливість залишати текстові коментарі до режимів обробки, ділитися досвідом та рекомендаціями щодо їх застосування.
- Додавання до обраного. Користувач може зберігати найбільш корисні режими у списку обраного для швидкого доступу в майбутньому.
- Пропонування нових режимів. Зареєстровані користувачі можуть пропонувати власні режими обробки для додавання до бази даних. Запропоновані режими проходять модерацію перед публікацією.
- Перегляд історії. Система зберігає історію пошуків користувача для зручності повторного доступу до раніше знайдених режимів.

Для адміністраторів:

- Модерація контенту. Адміністратор має можливість переглядати запропоновані користувачами режими та приймати рішення про їх публікацію або відхилення.
- Управління каталогами. Адміністратор може додавати, редагувати та видаляти записи у каталогах матеріалів та інструментів.
- Управління режимами. Адміністратор має повний доступ до бази даних режимів різання з можливістю додавання, редагування та видалення записів.
- Перегляд статистики. Система надає адміністратору статистичну інформацію про активність користувачів, популярність режимів та інші показники.

1.4.3 Нефункціональні вимоги

У межах проєктування програмного забезпечення сформовано комплекс нефункціональних вимог, що визначають якісні характеристики, експлуатаційні обмеження та критерії надійності функціонування системи

«CNC Helper». Специфікація вимог до доступності передбачає кросплатформність рішень із забезпеченням повноцінного функціонування інтерфейсу користувача через месенджер Telegram як на мобільних пристроях, так і на персональних комп'ютерах. Обов'язковою умовою є повна україномовна локалізація інтерфейсу, а також підтримка безперервного режиму експлуатації в режимі 24/7 із мінімізацією часових інтервалів технічного простою. Критерії продуктивності жорстко лімітують час відгуку системи на запити користувачів, який за нормальних умов функціонування та середнього мережевого навантаження не повинен перевищувати 3 секунди, при цьому архітектура обчислювальних модулів має гарантувати стабільну паралельну роботу щонайменше 100 одночасних активних сесій.

У розрізі інформаційної безпеки передбачено комплексний захист персональних та технологічних даних користувачів шляхом їхнього збереження у захищеному та структурованому вигляді. Доступ до інструментів адміністрування, модерації контенту та конфігурування бази даних суворо обмежений і надається виключно після проходження процедури верифікації та авторизації облікових записів, а програмний код містить вбудовані механізми захисту від поширених уразливостей та типових векторів атак, зокрема SQL-ін'єкцій та міжсайтового скриптингу (XSS). Вимоги до масштабованості системи реалізовано через проєктування гнучкої архітектури, здатної адаптуватися до зростання інтенсивності трафіку, та розробку реляційної структури бази даних, що дозволяє динамічно розширювати довідники інструментів і матеріалів без потреби рефакторингу вихідного коду. Наостанок, критерії підтримуваності програмного продукту забезпечуються високим рівнем документованості вихідного коду, його суворою відповідністю галузевим стандартам оформлення та використанням сучасного технологічного стеку, що володіє тривалим життєвим циклом та активною спільнотою профільної технічної підтримки.

1.4.4 Опис ролей користувачів

У системі визначено дві основні ролі: Користувач — основний користувач системи, оператор верстата з ЧПК або інший фахівець, що потребує інформації про режими різання. Користувач має доступ до всіх функцій пошуку, перегляду, оцінювання та коментування. Також може пропонувати нові режими для розгляду.

Адміністратор — користувач з розширеними правами, що відповідає за наповнення та актуальність бази даних. Адміністратор виконує модерацію користувацького контенту, управляє каталогами та має доступ до статистики системи.

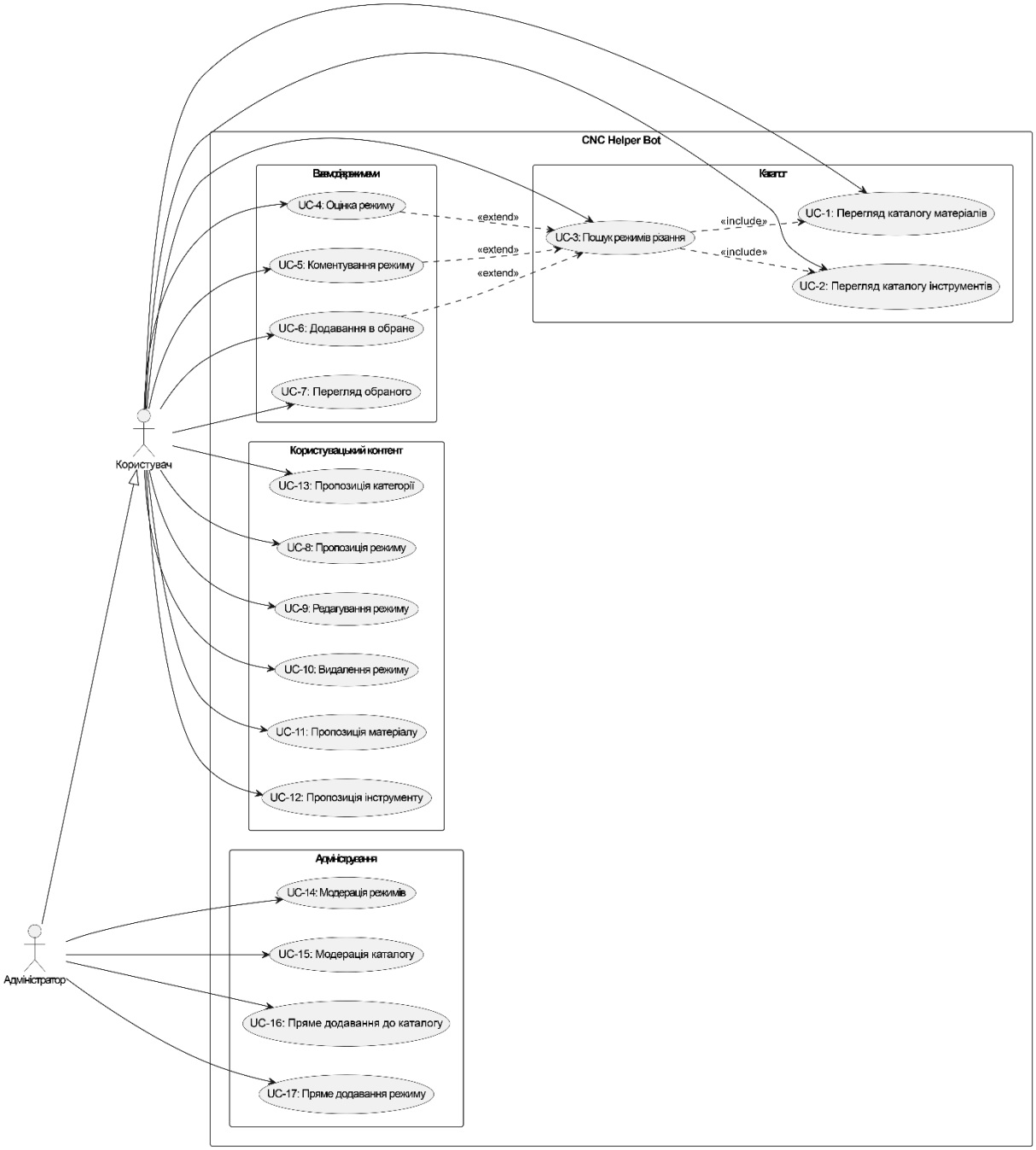


Рисунок *1.1 — Діаграма прецедентів системи «CNC Helper»*

Система передбачає розмежування прав доступу для забезпечення безпеки даних та зручності адміністрування. Основні сценарії взаємодії користувачів із системою, а також їхні ролі візуалізовано за допомогою діаграми прецедентів (див. рис. 1.1). На схемі відображено можливості

звичайного користувача щодо пошуку та збереження режимів, а також розширені функції адміністратора стосовно керування контентом та модерації

1.4.5 Структура даних

Для забезпечення логічної цілісності, збереження та ефективної обробки інформації в межах проєктованої системи визначено базовий перелік взаємопов'язаних сутностей та їхніх атрибутів, що формують структуру реляційної моделі даних. Сутність «Матеріал» (Material) призначена для класифікації оброблюваних заготовок і містить унікальний ідентифікатор, назву, категорію та текстовий опис із фізико-механічними характеристиками. Технологічні параметри різального оснащення акумулюються в сутності «Інструмент» (Tool), яка специфікується через унікальний ідентифікатор, назву й геометричний тип інструменту, його зовнішній діаметр і діаметр хвостовика, загальну довжину (OAL) та довжину різальної частини (LOC), а також кількість робочих зубів для фрезерного інструменту, матеріал виготовлення і тип нанесеного зносостійкого покриття. Ключовою сполучною ланкою системи є сутність «Режим різання» (CuttingMode), яка агрегує параметри обробки та містить унікальний ідентифікатор, зовнішні посилання на відповідні записи матеріалу й інструменту, розраховану частоту обертання шпинделя (RPM), швидкість подачі, глибину різання за один прохід, поточний статус запису в системі модерації (активний, на модерації або відхилений) та ідентифікатор автора для режимів, запропонованих користувачами.

Реєстрація та розмежування прав доступу реалізовані через сутність «Користувач» (User), де фіксуються унікальний внутрішній ідентифікатор, унікальний номер профілю в месенджері (Telegram ID), призначена системна роль (користувач або адміністратор) та тимчасова мітка дати реєстрації. Соціальна взаємодія, механізми оцінювання та накопичення емпіричного досвіду відображені через систему реляційних зв'язків між користувачами та режимами обробки. Сутність «Оцінка» (Vote) містить посилання на об'єкт режиму різання і профіль користувача з фіксацією бінарного типу оцінки у

вигляді схвалення чи несхвалення. Текстові відгуки формалізуються в сутності «Коментар» (Comment), яка поєднує посилання на режим різання і користувача з безпосереднім текстовим вмістом та датою створення запису. Наостанок, функціональний модуль персоналізації даних підтримується сутністю «Обране» (Favorite), що фіксує посилання на обраний користувачем режим різання, ідентифікатор його облікового запису та дату додавання елемента до персоналізованого списку швидкого доступу.

1.4.6 Технологічний стек

Для реалізації системи обрано наступний стек технологій:

- TypeScript — мова програмування з підтримкою статичної типізації;
- NestJS — фреймворк для створення серверних застосунків;
- grammY — бібліотека для розробки Telegram ботів;
- Prisma — ORM для роботи з базою даних;
- PostgreSQL — система управління базами даних;
- Docker — платформа для контейнеризації застосунку.

1.4.7 Завдання для реалізації

Для досягнення поставленої мети та забезпечення повноцінного функціонування інженерного сервісу визначено послідовність науково-технічних та прикладних завдань, що підлягають вирішенню в межах даної роботи. Першочерговим етапом є проєктування концептуальної архітектури системи з урахуванням принципів модульності, слабкого зв'язку компонентів та потенціалу горизонтального масштабування під навантаженням. На основі розробленої архітектури передбачено формування реляційної схеми бази даних, призначеної для структурованого зберігання інформації про конструкційні матеріали, геометрію інструментального оснащення, режими обробки, профілі користувачів та логіку їхньої соціальної взаємодії.

Практична реалізація інформаційного рівня вимагає створення автоматизованих міграцій бази даних та інтеграції об'єктно-реляційного

відображення за допомогою Prisma ORM. Наступний комплекс завдань охоплює безпосередню розробку програмних модулів у середовищі NestJS, що включає створення модуля взаємодії з Telegram Bot API на базі інструментарію бібліотеки grammY, розробку підсистеми ієрархічних каталогів для управління матеріалами та інструментами, а також реалізацію центрального обчислювального модуля режимів різання, оснащеного алгоритмами швидкого пошуку, бінарного оцінювання та текстового коментування даних.

Поряд із цим підлягає виконанню блок завдань щодо адміністрування та забезпечення безпеки, який передбачає розробку модуля користувачів із підтримкою гнучкого розмежування системних ролей та процедур авторизації, реалізацію сервісу персоналізації для збереження обраних технологічних режимів, а також проєктування багаторівневого модуля модерації з метою верифікації та обробки вхідного користувацького контенту. Фінальна стадія дослідження передбачає проведення комплексного функціонального та навантажувального тестування системи, усунення виявлених програмних дефектів і девіацій, а також підготовку детальної технічної документації для подальшого розгортання, конфігурування та експлуатації створеного програмного продукту. Реалізація зазначеного комплексу завдань спрямована на створення надійного, відмовостійкого та ергономічного цифрового сервісу, здатного оптимізувати виробничу діяльність операторів верстатів з ЧПК, мінімізувати ймовірність виникнення технологічних помилок при виборі параметрів обробки та забезпечити ефективне середовище для акумуляції й обміну практичним інженерним досвідом між фахівцями машинобудівної галузі.

2. ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Аналіз функціональних та нефункціональних вимог

Процес проєктування системи «CNC Helper» розпочався з детального збору та формалізації вимог, що висуваються до сучасних довідкових систем у сфері виробництва. Оскільки основним завданням системи є надання точних технологічних параметрів у максимально короткий термін, вимоги було розділено на дві ключові категорії.

2.1.1 Функціональні вимоги та ролі користувачів

З метою забезпечення повноцінного функціонування та життєдіяльності програмного комплексу визначено дві базові ролі користувачів, кожна з яких наділена специфічним набором привілеїв та функціональних можливостей. Перша роль представлена користувачем або оператором, для якого критично важливим експлуатаційним фактором є висока швидкість доступу до технологічних даних безпосередньо в умовах виробничого процесу. Специфікація функціональних вимог для даної категорії акторів передбачає можливість оперативної навігації за допомогою деревоподібного каталогу конструкційних матеріалів та різальних інструментів, а також застосування системи динамічних фільтрів для точного підбору регламентованих режимів різання. Крім того, для верифікації розрахованих параметрів обробки інтегровано інструменти соціальної взаємодії у вигляді системи бінарних оцінок та текстових коментарів, реалізовано механізм персоналізації інженерного досвіду через спеціалізований розділ обраного, а також розроблено програмний інструментарій для самостійного розширення інформаційної бази шляхом надсилання пропозицій щодо створення нових записів, що трансформує архітектуру сервісу в децентралізовану краудсорсингову платформу. Друга роль визначається як адміністратор або модератор, на якого покладено функції забезпечення логічної цілісності,

актуальності та загальної достовірності накопичуваних технологічних даних. Комплекс вимог до даної ролі охоплює інструменти оперативного управління чергою модерації для перевірки надісланого користувачами контенту, можливість адміністрування та редагування ієрархічної структури каталогів, включаючи генерацію нових категорій, а також функціонал автоматизованого сповіщення користувачів про поточний статус розгляду їхніх технологічних запитів.

2.1.2 Нефункціональні вимоги

Окрім безпосереднього виконання цільових функцій, до проєктованого програмного комплексу висунуто ряд системних вимог, що гарантують його загальну стабільність, відмовостійкість та експлуатаційну надійність. Першочерговим пріоритетом у цьому контексті є продуктивність системи, яка регламентує, що час відгуку чат-бота на технологічний запит користувача не повинен перевищувати інтервал в одну-дві секунди. Зазначений показник швидкодії забезпечується шляхом глибокої оптимізації індексування та побудови реляційних запитів до системи управління базами даних PostgreSQL. Не менш важливим критерієм є доступність сервісу, архітектура якого має гарантувати безперервне функціонування в режимі 24/7, що обумовлено специфікою сучасних машинобудівних підприємств, де виробничі процеси обробки деталей на верстатах з ЧПК часто мають циклічний або цілодобовий характер. Наостанок, критерій надійності покладається на інтегровані механізми об'єктно-реляційного відображення Prisma ORM, які на рівні програмної логіки забезпечують сувору цілісність зв’язків між сутностями, повністю унеможливлюючи виникнення неузгоджених або ізольованих записів у структурі бази даних при використанні каскадних операцій видалення чи оновлення інформації.

2.2 Обґрунтування вибору технологічного стеку

Вибір інструментарію для розробки «CNC Helper» базувався на принципах модульності, типізації та швидкості розгортання.

2.2.1 TypeScript як мова розробки

На відміну від стандартного JavaScript, TypeScript надає можливість суворої типізації даних. У контексті розробки довідника режимів різання, де оперують числовими параметрами (оберти, подача), використання інтерфейсів та типів дозволяє уникнути цілого класу помилок ще на етапі написання коду. Це гарантує, що розрахункові модулі завжди отримуватимуть дані у правильному форматі.

2.2.2 Фреймворк NestJS та архітектурний патерн DI

NestJS було обрано завдяки його структурованості. Фреймворк нав’язує використання модульної архітектури, що ідеально підходить для диплому, оскільки дозволяє чітко розділити логіку бота, логіку бази даних та сервіси модерації. Використання Dependency Injection (впровадження залежностей) спрощує тестування та подальше розширення системи.

2.2.3 Бібліотека grammY та Telegram Bot API

Для реалізації інтерфейсу було обрано grammY — сучасний та високопродуктивний фреймворк для Telegram-ботів. Його перевага полягає у нативній підтримці TypeScript та ефективній роботі з Middleware, що дозволяє легко реалізувати перевірку прав доступу (адмін/користувач) для кожного повідомлення.

2.3 Проєктування структури бази даних

Схема бази даних розроблена з урахуванням нормалізації для усунення надмірності даних та забезпечення їхньої цілісності (див. рис. 2.1).

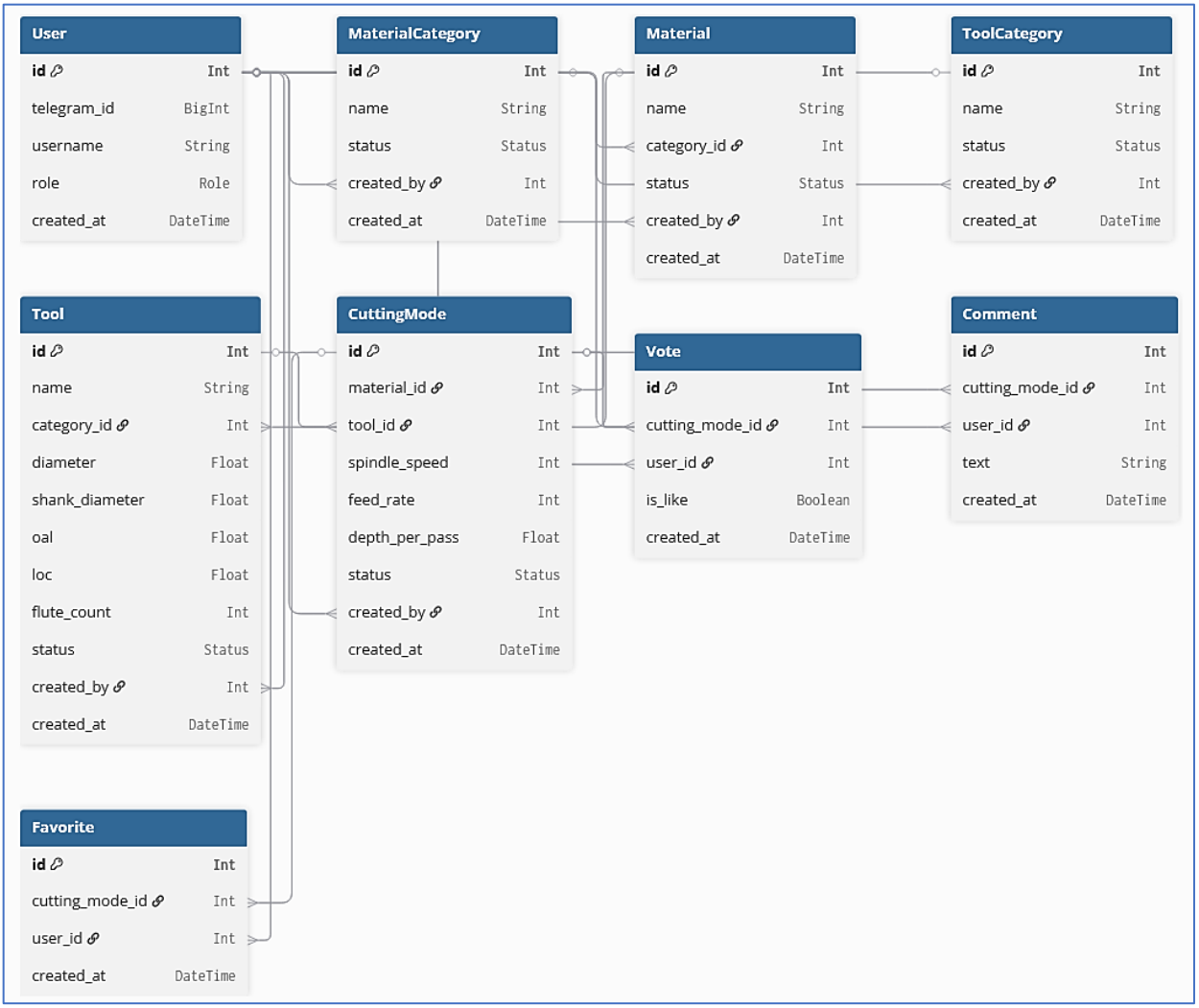


Рисунок *2.1 — Схема бази даних*

Суб'єкти та об'єкти проєктованої інформаційної системи структуровані у межах реляційної моделі даних та представлені набором взаємопов'язаних таблиць, кожна з яких виконує специфічні функції щодо збереження та обробки інформації. Первинним ідентифікатором для генерації користувацького контенту, реалізації механік голосування та формування списків швидкого доступу є таблиця «User», яка забезпечує фіксацію

унікального ідентифікатора месенджера Telegram ID із застосуванням типу даних BigInt, імені користувача та його поточної системної ролі (USER або ADMIN). Ієрархічна класифікація та структурування довідників реалізовані через сутності верхнього рівня «MaterialCategory» та «ToolCategory», що забезпечують категоризацію технологічних об'єктів. Безпосередні дані про фізико-механічні властивості оброблюваних заготовок акумулюються в таблиці «Material», яка пов'язана з відповідною категорією за допомогою обмежень зовнішнього ключа. Геометричні та конструктивні параметри різального оснащення описуються в таблиці «Tool», де реєструються діаметр різальної частини, діаметр хвостовика, загальна довжина (OAL), робоча довжина різальної кромки (LOC), а також кількість робочих зубів інструменту.

Центральною асоціативною сутністю бази даних є таблиця «CuttingMode», яка формалізує реляційний зв'язок між конкретним конструкційним матеріалом та застосовуваним інструментом. Дана таблиця агрегує розраховані технологічні параметри механічної обробки, такі як швидкість обертання шпинделя, швидкість подачі та глибина різання за один прохід, а також маркується відповідним системним статусом у межах підсистеми адміністрування (PENDING, APPROVED, REJECTED). Соціальна взаємодія, процеси краудсорсингу та функції персоналізації досвіду операторів реалізовані через транзакційні таблиці «Vote», «Comment» та «Favorite», що відображають взаємозв'язки користувачів із доступними режимами обробки. Ці зв'язки жорстко регламентуються обмеженнями унікальності на рівні системи управління базами даних, що, наприклад, унеможливлює повторне голосування одного й того самого користувача за ідентичний технологічний режим. Крім того, інтеграція механізмів каскадного видалення (onDelete: Cascade) на рівні декларативного опису схеми бази даних гарантує абсолютну логічну цілісність інформаційного фонду: у разі видалення адміністратором певного матеріалу або інструменту, всі деривативні зв'язані записи режимів різання, коментарів та оцінок

автоматично вилучаються або архівуються СУБД, повністю виключаючи виникнення неузгоджених даних.

2.4 Проєктування функціоналу та моделі взаємодії

Програмний комплекс «CNC Helper» базується на моделі управління доступом на основі ролей (Role-Based Access Control — RBAC), що є фундаментом для забезпечення безпеки та цілісності даних у багатокористувацьких системах. Проєктування логіки взаємодії було спрямоване на мінімізацію когнітивного навантаження на оператора верстата, який часто працює в умовах обмеженого часу та зашумленого виробничого середовища.

2.4.1 Архітектура взаємодії як скінченний автомат

Процес пошуку та отримання оптимальних технологічних параметрів механічної обробки в проєктованій інформаційній системі реалізовано на основі математичної моделі скінченного автомата. З метою мінімізації ймовірності виникнення синтаксичних помилок та девіацій, зумовлених варіативністю ручного текстового введення назв конструкційних матеріалів чи геометричних конфігурацій інструментального оснащення, взаємодію з користувачем побудовано у вигляді чітко детермінованої послідовності кроків. На першому етапі здійснюється селекція об'єкта обробки, при якій користувач спочатку обирає узагальнену ієрархічну категорію, після чого система генерує релевантний відфільтрований перелік конкретних матеріалів. Після однозначної ідентифікації матеріалу заготовки алгоритм переходить до другого етапу, що передбачає вибір типу різального інструменту та визначення його базових фізико-геометричних характеристик. На фінальній стадії скінченного автомата система виконує агрегування та аналіз відповідних даних з асоціативної таблиці режимів різання, трансформуючи та виводячи підраховані параметри у структурованому, зручному для сприйняття

інженерним персоналом вигляді. Описаний покроковий підхід гарантує, що кожна транзакція та пошуковий запит до бази даних містять виключно валідні системні ідентифікатори, що суттєво знижує обчислювальне навантаження на сервер та пришвидшує процес обробки інформації інформаційною системою.

2.4.2 Модель життєвого циклу контенту та модерація

Важливою складовою архітектурного проєктування інформаційної системи є логіка наповнення та верифікації реляційної бази даних. Оскільки в основу функціонування платформи закладено децентралізовану краудсорсингову модель, що передбачає активну взаємодію та взаємодопомогу в межах інженерної спільноти, було розроблено та інтегровано спеціалізований механізм відкладеної публікації даних. Згідно з цим алгоритмом, усі нові технологічні пропозиції щодо режимів обробки, які надходять від операторів, за замовчуванням маркуються системним прапором PENDING, що автоматично ініціює детермінований ланцюжок подій. На першому етапі внесений запис успішно персистується у базі даних, проте програмно ізолюється і не бере участі у формуванні результатів загального пошуку для інших користувачів з метою запобігання поширенню неперевірених параметрів. Одночасно з цим підсистема адміністрування генерує та надсилає релевантне сповіщення про надходження нової заявки у спеціалізований інтерфейс модерації. Після проведення експертної оцінки та верифікації параметрів, яка включає обов'язкову перевірку на відповідність фізико-механічним закономірностям процесу різання та технологічним можливостям обладнання, адміністратор змінює статус запису на APPROVED. Ця дія викликає фінальну тригерну процедуру, за якої режим інтегрується у глобальну пошукову базу, а система в автоматичному режимі надсилає автору інформаційне сповіщення про успішне внесення його технологічного внеску до загального реєстру.

2.4.3 Оптимізація представлення даних та пагінація

З огляду на обмежені можливості відображення великих обсягів тексту в інтерфейсі месенджера Telegram, було спроєктовано систему пагінації. При великій кількості знайдених варіантів (наприклад, багато різних режимів для одного інструменту), бот видає дані порційно. Це реалізовано через передачу параметрів limit та offset у запитах до Prisma ORM, що дозволяє динамічно перемикати сторінки результатів без повторного завантаження всієї вибірки з пам'яті.

2.4.4 Механіка соціального рейтингу та UPSERT-логіка

Для забезпечення високої достовірності та актуальності бази даних режимів різання системи «CNC Helper» спроєктовано комплексну підсистему соціального підтвердження технологічних параметрів. В умовах реального машинобудівного виробництва, де девіації та помилки у виборі режимів обробки призводять до виходу з ладу високовартісного інструментального оснащення або незворотного псування заготовок, механізми верифікації інформаційного фонду через акумуляцію колективного емпіричного досвіду користувачів набувають критично важливого значення. Концептуальне проєктування зазначеної моделі соціальної взаємодії базується на детермінованому принципі бінарного обмеження, що передбачає можливість генерації лише однієї оцінки від одного унікального користувача для конкретного технологічного запису, завдяки чому формується об'єктивний і захищений від штучних маніпуляцій рейтинг довіри до режимів обробки.

Функціональне ядро та логічна архітектура даної підсистеми інтегрують три фундаментальні аспекти, пов'язані з ідентифікацією користувачів, обробкою транзакційних станів та оптимізацією обчислювальних ресурсів. У розрізі ідентифікації та забезпечення унікальності голосу програмні алгоритми заблоковують спроби дубльованого оцінювання від одного актора шляхом застосування механізму атомарного оновлення існуючого реляційного відношення замість накопичення надлишкових записів. Це

дозволяє оператору оперативно актуалізувати свій технологічний відгук, зокрема реверсувати позитивну оцінку на негативну за результатами тривалих натурних випробувань матеріалу на верстаті з ЧПК, не порушуючи при цьому загальну статистичну та структурну цілісність бази даних.

Процес динамічної обробки станів оцінювання базується на трирівневій моделі взаємодії, яка включає позитивну детермінацію (фіксацію схвалення), негативну детермінацію (фіксацію несхвалення) та нейтральний стан, що відповідає процедурі легітимного скасування раніше поданого голосу. Зазначена трирівнева архітектура забезпечує гнучке управління зворотним зв'язком з боку інженерного персоналу, що безпосередньо підвищує дескриптивну точність результуючого рейтингу системи. З технічного погляду, використання логіки модифікації існуючих записів на етапі проєктування сховища даних дозволяє мінімізувати навантаження на дискову підсистему пам’яті, знизити рівень фрагментації таблиць та суттєво пришвидшити виконання складних агрегуючих SQL-запитів при розрахунку інтегральних індексів довіри в режимі реального часу. Описаний науково-практичний підхід до формування механіки соціального верифікування трансформує розроблену систему зі стаційного інформаційного довідника в динамічне середовище обміну професійними інженерними знаннями, де кожен доступний запис володіє підтвердженою практичною цінністю для виробництва.

2.5 Висновки до розділу 2

У межах другого розділу здійснено комплексне проєктування інформаційної системи «CNC Helper», результати якого базуються на системному аналізі предметної області та технічних умов подальшої експлуатації програмного продукту. На основі проведеного дослідження формалізовано повний перелік функціональних і нефункціональних вимог, що забезпечило чітке окреслення меж проєкту та детермінацію таких

пріоритетних критеріїв розробки, як висока швидкість пошуку, надійність збереження інформації та ергономічність користувацького інтерфейсу. Науково обґрунтовано вибір оптимального технологічного стеку, де синергія мови програмування TypeScript, фреймворку NestJS та інструментарію Prisma ORM визначена як найбільш ефективне рішення для створення суворо типізованого, масштабованого та рефакторопридатного серверного застосунку.

У розрізі проєктування інформаційного рівня розроблено реляційну архітектуру бази даних у середовищі системи управління базами даних PostgreSQL, яка через механізми обмежень зовнішніх ключів та індексування гарантує логічну цілісність нормативно-довідкової технологічної інформації та оптимізує процеси збереження динамічного користувацького контенту. Крім того, спроєктовано поведінкову модель взаємодії компонентів на основі концепції скінченних автоматів та дворівневої рольової моделі доступу, що в сукупності мінімізує ймовірність виникнення синтаксичних і технологічних помилок при виборі параметрів обробки, а також формує захищене середовище для модерації даних та акумуляції колективного емпіричного досвіду операторів верстатів з ЧПК. Викладені теоретичні та конструктивні результати проєктування слугують безпосереднім методологічним підґрунтям та інженерною інструкцією для етапу програмної реалізації системи, аналіз якої наводиться у наступній частині кваліфікаційної роботи.

3. РОЗРОБКА ТА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Структура проєкту та налаштування середовища

Розробка серверної частини Telegram-бота базується на платформі Node.js, яка забезпечує асинхронне виконання коду, що є критично важливим для обробки великої кількості одночасних запитів від користувачів месенджера. В якості архітектурного фундаменту використовується фреймворк NestJS. Процес ініціалізації проєкту розпочинається зі створення базової структури директорій за допомогою інтерфейсу командного рядка (CLI) NestJS.

Коренева структура проєкту містить директорію src/, в якій розташовано головні файли ініціалізації: main.ts та app.module.ts. Файл main.ts виступає точкою входу в застосунок. У ньому викликається метод NestFactory.create(), який компілює дерево залежностей та запускає життєвий цикл програми.

Основою архітектури NestJS є модульність. Файл app.module.ts виконує роль кореневого модуля, який імпортує всі інші дочірні функціональні модулі системи. Такий підхід гарантує жорстку інкапсуляцію: кожен компонент відповідає виключно за власну предметну область, що полегшує підтримку та масштабування кодової бази. Розподіл відповідальності між основними модулями системи наведено у Таблиці 3.1.

Для кожного модуля створюється власна директорія, яка містить контролери (для обробки вхідних запитів) та сервіси (для реалізації бізнес-логіки). Окрім сервісів, у директоріях модулів розташовуються додаткові компоненти: наприклад, для BotModule реалізовано обробники команд у піддиректорії handlers/, клавіатури у keyboards/ та проміжні програмні забезпечення у middleware/. Загальну структуру директорій та файлів розробленого застосунку наведено на рисунку 3.1.

Таблиця *3.1 — Відповідальність модулів NestJS у структурі застосунку*

Модуль	Відповідальність
BotModule	Ініціалізація grammY, обробка команд та callback-кнопок, middleware (авторизація, визначення ролі)
UserModule	Реєстрація за Telegram ID, управління ролями
CatalogModule	CRUD для матеріалів, інструментів та їх категорій
CuttingModeModule	Пошук режимів, створення, голосування, коментарі
FavoriteModule	Додавання / видалення / перегляд обраного
ModerationModule	Черга модерації, схвалення / відхилення, сповіщення
PrismaModule	Підключення до БД, глобальний сервіс Prisma

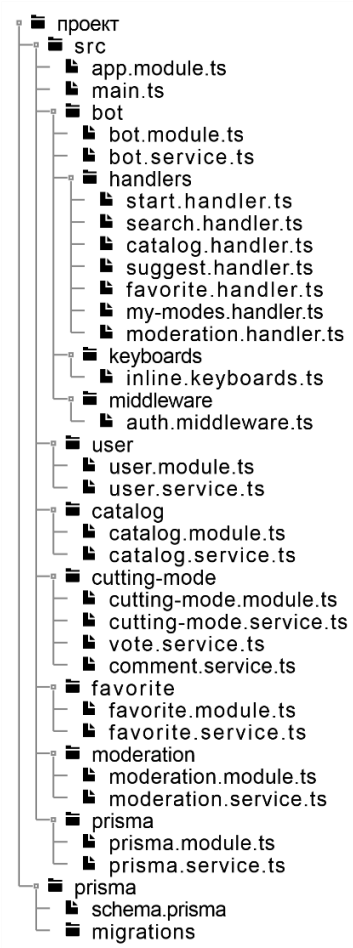


Рисунок *3.1 — Ієрархічна структура директорій та файлів застосунку*

3.2 Розробка модуля доступу до даних на основі Prisma ORM

Центральним компонентом архітектури збереження даних у системі CNC Helper є рівень доступу до даних (Data Access Layer), реалізований за допомогою сучасної ORM-системи Prisma. Вибір реляційної моделі даних та СУБД PostgreSQL обумовлений необхідністю забезпечення суворої цілісності інформації про режими різання, де кожна помилка в типах даних або зв’язках може призвести до некоректних обчислень на виробництві.

3.2.1 Конфігурація схеми даних та генерація клієнта

Процес розробки розпочався з формування декларативного опису структури бази даних у файлі schema.prisma. На відміну від традиційного підходу написання SQL-запитів, Prisma дозволяє описувати сутності мовою моделювання даних (Data Modeling Language — DML). Кожна таблиця бази даних представлена у вигляді моделі, де чітко визначені типи полів, модифікатори доступу (наприклад, @unique для Telegram ID) та значення за замовчуванням (наприклад, часові мітки @default(now())).

Після опису моделей виконується процедура генерації Prisma Client — спеціалізованого пакета, який створює об’єктно-орієнтований інтерфейс для роботи з базою. Це критично важливий етап, оскільки він забезпечує повну типізацію на етапі компіляції. Наприклад, спроба звернутися до поля, якого не існує в таблиці Tool, буде виявлена компілятором TypeScript ще до запуску коду, що нівелює ризик виникнення помилок у середовищі виконання.

3.2.2 Реалізація реляційних зв'язків та обмежень цілісності

Особлива увага в процесі проєктування та реалізації інформаційного рівня системи була приділена конфігуруванню та оптимізації реляційних зв'язків між структурними таблицями бази даних. В архітектурі розробленого програмного комплексу реалізовано кілька типів відношень, які забезпечують логічну відповідність даних предметній області. Зокрема, зв'язок типу «один-

до-багатьох» (1:N) застосовано для формалізації авторства технологічного контенту, де один зареєстрований користувач володіє можливістю генерації множини режимів різання, тоді як кожен окремий режим однозначно ідентифікується з конкретним автором за допомогою обмежень зовнішніх ключів created_by у відповідних таблицях. Відношення типу «багато-до-багатьох» (M:N) інтегровано для забезпечення функціональності модулів персоналізації «Обране» (Favorite) та соціального оцінювання «Голосування» (Vote). При цьому інструментарій Prisma ORM здійснює автоматизоване керування проміжними сполучними таблицями на рівні бази даних, що дозволяє оперувати високорівневими методами вибірки та маніпулювання даними без потреби ручного опису складних і обчислювально містких операцій JOIN.

З метою гарантування високої надійності, відмовостійкості та актуальності інформаційного фонду на рівні системи управління базами даних впроваджено механізми каскадного видалення записів (onDelete: Cascade). Застосування цього підходу детермінує, що в разі вилучення з бази даних облікового запису користувача, СУБД в автоматичному режимі здійснює транзакційне видалення всіх деривативних пов'язаних сутностей, включаючи його коментарі та подані голоси. Описане інженерне рішення повністю запобігає прогресуючому накопиченню надлишкових, несумісних або ізольованих фрагментів інформації та виключає порушення логічної цілісності сховища через виникнення так званих сирітських записів (orphan records).

3.2.3 Імплементація PrismaService в екосистему NestJS

Для інтеграції доступу до даних у модульну структуру NestJS було розроблено PrismaModule. В основі модуля лежить сервіс PrismaService, який розширює базовий клас PrismaClient. Цей сервіс виконує роль «синглтона» — єдиного екземпляра з’єднання, який впроваджується (Dependency Injection) в інші модулі.

У лістингу коду сервісу реалізовано методи життєвого циклу onModuleInit, які забезпечують автоматичне відкриття з’єднання з PostgreSQL при запуску сервера. Такий підхід дозволяє уникнути затримок (cold start) при першому запиті користувача до бота. Крім того, реалізовано логування запитів у режимі розробки, що дозволяє аналізувати ефективність SQL-запитів, які генеруються ORM-системою, та вчасно оптимізувати індекси.

3.2.4 Управління станом бази даних через міграції

Невід’ємною частиною розробки модуля даних є система міграцій Prisma Migrate. Кожна зміна в моделях фіксується у вигляді окремого SQL-файлу з часовою міткою. Це дозволяє команді розробників синхронізувати структуру бази даних та безпечно розгортати зміни на сервері. Для дипломного проєкту це також є гарантією відтворюваності середовища: за допомогою однієї команди npx prisma migrate deploy можна підготувати порожню базу даних до роботи, автоматично створивши всі необхідні таблиці та обмеження згідно з проєктною документацією.

3.3 Реалізація інтерфейсу користувача та бізнес-логіки Telegram-бота

Розробка інтерфейсу системи «CNC Helper» суттєво відрізняється від класичних графічних застосунків (GUI), оскільки взаємодія з користувачем відбувається в межах обмеженого простору месенджера Telegram. Основним інструментом побудови інтерфейсу виступає бібліотека grammY, яка інтегрується в екосистему NestJS як окремий модуль BotModule.

3.3.1 Архітектура обробки повідомлень та Middleware

Взаємодія чат-бота з сервером Telegram побудована на принципах подійно-орієнтованого та реактивного програмування, де кожна дія користувача, включаючи надсилання системних команд, текстових

повідомлень або активацію інлайн-кнопок, генерує об'єкт події типу Update, який транслюється на сервер застосунку. З метою забезпечення інформаційної безпеки, авторизації та персоналізації досвіду операторів безпосередньо на рівні BotModule інтегровано архітектурний шар проміжного програмного забезпечення (Middleware). Елементи користувацького інтерфейсу, що забезпечують взаємодію з архітектурними шарами застосунку, динамічно адаптуються під роль авторизованого актора. Після успішної ініціалізації системи командою /start користувачеві генерується стартове вікно навігації (див. рис. 3.2).



Рисунок *3.2 — Стартове вікно чат-бота «CNC Helper»*

Перед передачею вхідного запиту на рівень контролерів бізнес-логіки, кожна транзакція проходить послідовну фільтрацію через три детерміновані етапи, що включають ідентифікацію користувача, верифікацію прав доступу та системне логування подій. На етапі ідентифікації програмні алгоритми

здійснюють перевірку наявності унікального ідентифікатора telegram_id у реляційній базі даних, і в разі виявлення нового актора спеціалізований сервіс UserService в автоматичному режимі виконує його реєстрацію з призначенням базової ролі за замовчуванням USER. Далі механізм визначення прав доступу зчитує актуальну системну роль користувача зі сховища даних та імплементує її в об'єкт контексту запиту (ctx), що дозволяє динамічно керувати інтерфейсом чат-бота, відображаючи або приховуючи спеціалізовані кнопки адміністративного меню модерації. Фінальний превентивний етап передбачає обов'язкове логування кожної взаємодії, що забезпечує фіксацію телеметрії для подальшого аналізу користувацької активності, моніторингу навантаження та оперативного виявлення потенційних девіацій у логіці навігації скінченного автомата. Описаний процес обробки вхідних інформаційних пакетів базується на класичному патерні проєктування ланцюжка обов'язків, де запит послідовно передається через стек зв'язаних компонентів. Загальну архітектурну схему взаємодії зовнішнього інструментарію Telegram API, фреймворку бібліотеки grammY та внутрішніх сервісів NestJS зображено на схемі (див. рис. 3.3).

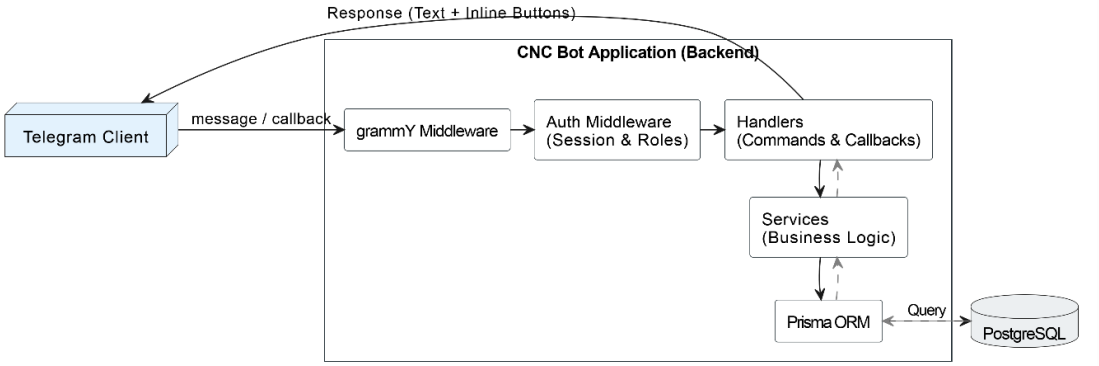


Рисунок *3.3 — Схема взаємодії компонентів системи при обробці запитів.*

3.3.2 Механіка динамічних Inline-клавіатур

Основною складністю при розробці було створення зручної навігації каталогом, який містить велику кількість категорій, матеріалів та

інструментів. Для вирішення цього завдання було розроблено компонент InlineKeyboardBuilder. На відміну від статичних меню, клавіатури в «CNC Helper» генеруються динамічно на основі актуального стану бази даних. При запиті списку матеріалів сервіс виконує звернення до PrismaService, отримує масив об'єктів і трансформує їх у сітку кнопок. Кожна кнопка містить callback_data — унікальний рядок, що несе в собі ідентифікатор об'єкта та код дії. Це дозволяє системі точно знати, який саме елемент обрав користувач, без необхідності зберігати стан сесії на стороні сервера у складних структурах.

3.3.3 Реалізація покрокового сценарію пошуку режимів

Сценарій пошуку та отримання технологічних параметрів є найбільш критичним процесом інформаційної системи, оскільки він безпосередньо детермінує швидкість доступу оператора верстата з ЧПК до верифікованих результатів обробки. Програмна реалізація даного процесу базується на математичній моделі скінченного автомата (State Machine), у межах якої користувач послідовно проходить через систему детермінованих ієрархічних фільтрів. Початковий етап передбачає вибір об'єкта обробки, при якому оператор після ініціалізації інтерфейсу переходить до загального каталогу матеріалів, де визначає узагальнену класифікаційну категорію (див. рис. 3.4).

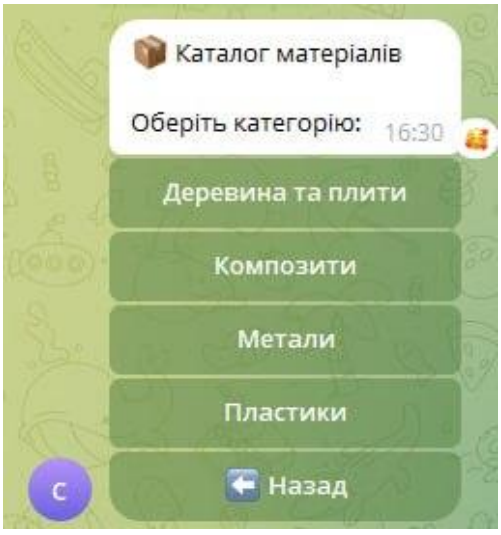


Рисунок *3.4 — Вікно вибору категорії конструкційних матеріалів*

Після вибору категорії система генерує відфільтрований перелік, у якому користувач вказує конкретний матеріал заготовки (див. рис. 3.5).

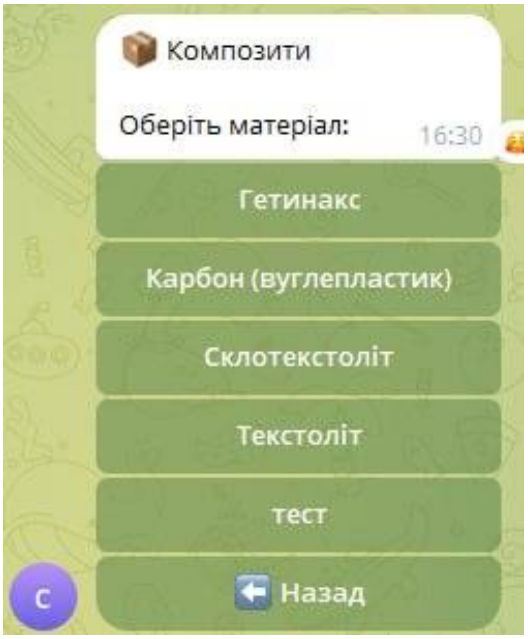


Рисунок *3.5 — Вікно вибору конкретного об'єкта обробки*

Після однозначної ідентифікації матеріалу автомат переходить до етапу селекції різального інструменту, пропонуючи користувачеві вказати категорію інструментального оснащення, релевантного для обраної фізико-механічної групи (див. рис. 3.6).

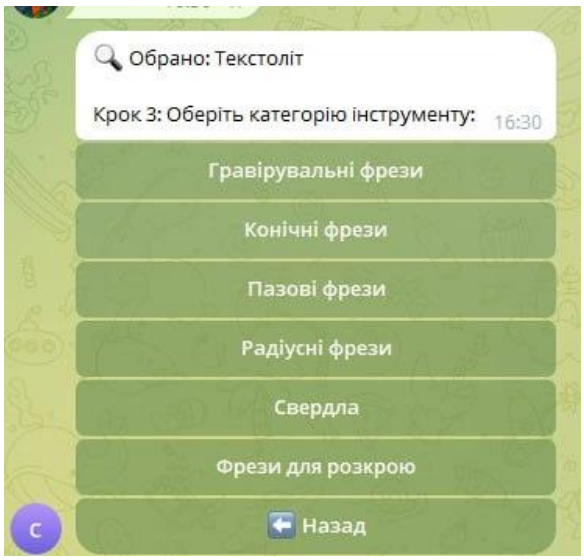


Рисунок *3.6 — Меню вибору класифікаційної категорії інструменту*

На основі обраної категорії фрез система формує деталізований перелік моделей із фіксацією їхніх геометричних характеристик, зокрема номінального діаметра та кількості робочих зубів (див. рис. 3.7).



Рисунок *3.7 — Список інструментів із зазначенням конструктивних параметрів*

На фінальній стадії скінченного автомата на основі сформованої пари суворих системних ідентифікаторів виконується транзакційний запит до

асоціативної таблиці CuttingMode з подальшим виведенням розрахованих параметрів обробки. Ключовою технічною та архітектурною особливістю реалізації описуваного сценарію є використання стандартизованого механізму Callback Data в межах Telegram Bot API. Оскільки специфікація месенджера встановлює жорстке лімітування обсягу метаданих для кожної інлайн-кнопки на рівні 64 байтів, було спроєктовано та впроваджено компактний протокол серіалізації даних у форматі структурованого рядка. Кожен керуючий елемент інтерфейсу містить префікс цільової дії та унікальний числовий ідентифікатор об'єкта, що дозволяє серверу застосунку миттєво інтерпретувати контекст натискання кнопки без потреби збереження та синхронізації складних об'єктів сесій у оперативній пам'яті, забезпечуючи реалізацію концепції безстанової (State-less) архітектури. З метою оптимізації інтерфейсу користувача та підвищення ергономічності взаємодії в системі реалізовано підсистему динамічної пагінації даних. У випадках, коли для обраної пари матеріалу та інструментального оснащення у базі даних виявлено множину альтернативних режимів різання, чат-бот обмежує одночасне виведення масиву інформації для запобігання перевантаженню екрана мобільного пристрою. Замість цього алгоритм генерує уніфіковану інформаційну картку одного режиму, оснащену інтерактивними навігаційними елементами горизонтального гортання. При активації зазначених кнопок переходу програмний код виконує не деструктивне перестворення повідомлення, а транзакційне редагування його текстового вмісту за допомогою методу editMessageText, що забезпечує візуальну плавність зміни інтерфейсних станів та мінімізує обсяги мережевого трафіку (див. рис. 3.8).

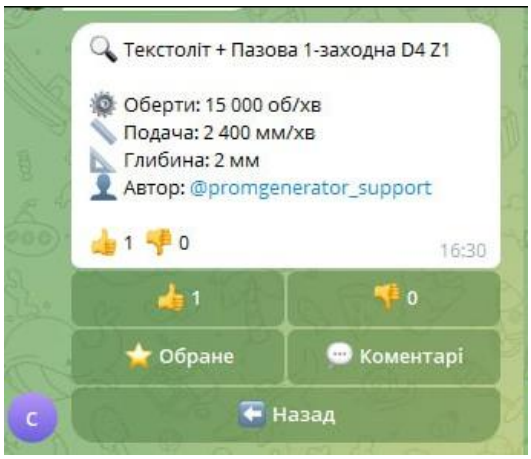


Рисунок *3.8 — Картка видачі розрахованого режиму різання з активованими індикаторами*

Користувацький шлях під час пошуку режиму різання реалізовано як послідовність вибору параметрів. Детальний життєвий цикл запиту від початкової команди до отримання фінального результату з бази даних відображено на діаграмі послідовності (див. рис. 3.9).

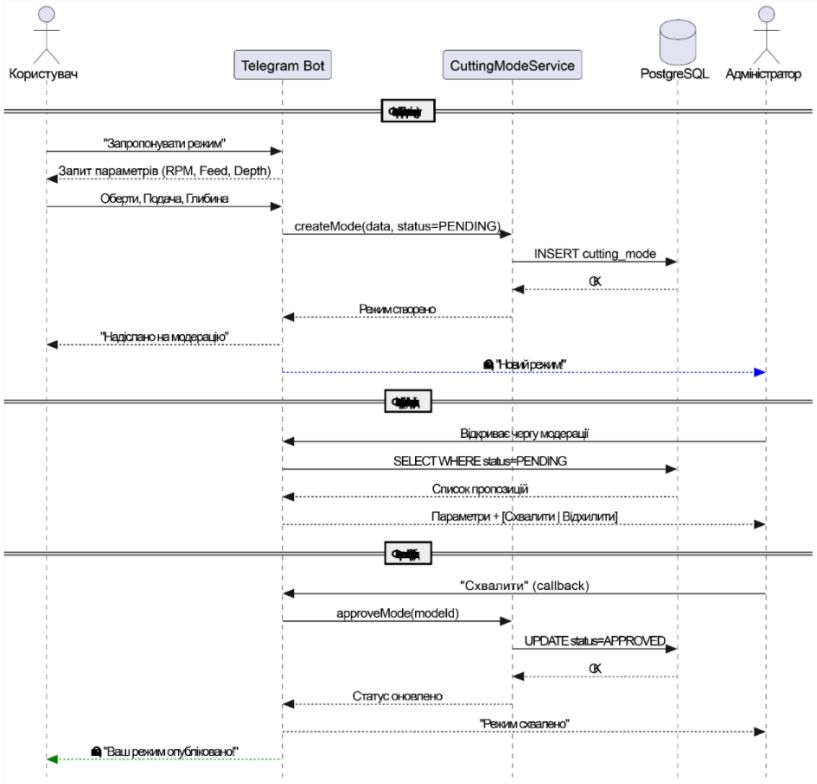


Рисунок *3.9 — Діаграма послідовності процесу пошуку параметрів обробки*

3.3.4 Система соціальної взаємодії: рейтинги та коментарі

З метою підвищення рівня достовірності та актуальності нормативно-довідкової інформації в системі реалізовано програмну механіку інтерактивного зворотного зв'язку, інтегровану безпосередньо в інтерфейс виведення технологічних параметрів. Під кожною інформаційною карткою знайденого режиму обробки сформовано блок керуючих інлайн-кнопок, призначених для бінарного оцінювання результатів.

Приклад початкової генерації інтерфейсної картки з технологічними параметрами обробки (частота обертання, швидкість подачі, глибина різання) та нульовими індикаторами соціального відгуку наведено на рисунку 3.10

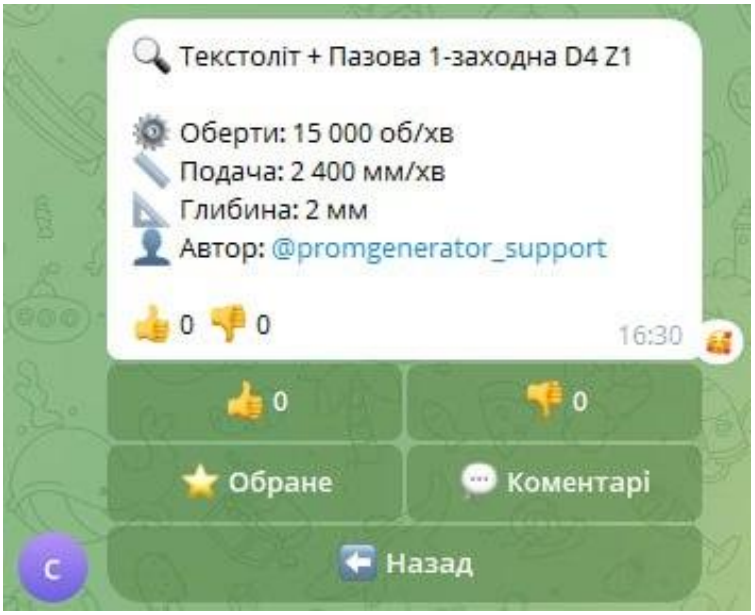


Рисунок *3.10 — Результат пошуку з початковим станом блоку оцінювання*

При активації користувачем елемента оцінювання спеціалізований сервіс VoteService ініціює транзакційну перевірку наявності релевантних записів про голосування даного актора у реляційній базі даних, яка функціонує за триваріантним алгоритмом обробки станів. У випадку відсутності попередньої активності поточна оцінка персистується у сховищі як новий запис. Якщо користувач повторно активує ідентичний елемент інтерфейсу,

система інтерпретує цю дію як скасування зворотного зв'язку та анулює раніше поданий голос. У сценарії, коли оператор змінює технологічне рішення, програмний код виконує атомарну модифікацію відношення, за якої попередній запис деструктується, а нова детермінована оцінка успішно реєструється в базі даних. Описана логіка жорстко гарантує об'єктивність результуючого рейтингу за принципом обмеження одного голосу від одного унікального профілю.

Для забезпечення можливості текстового рецензування та обміну практичним досвідом під карткою режиму інтегровано модуль коментування. Натискання відповідної інлайн-кнопки переводить користувача до вікна перегляду наявних відгуків спільноти та активації форми введення нового повідомлення, як продемонстровано на рисунку 3.11.

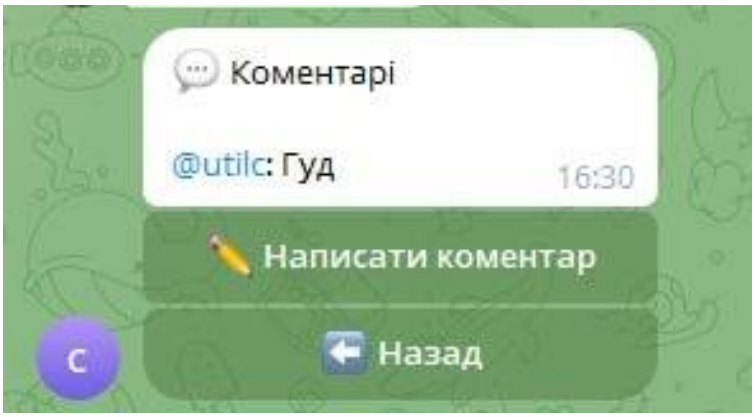


Рисунок *3.11 — Інтерфейс перегляду та додавання текстових коментарів*

Такий підхід дозволяє формувати високий рівень довіри до технологічних параметрів, які отримали всебічне практичне підтвердження від професійної спільноти операторів верстатів з ЧПК.

3.4 Реалізація підсистеми модерації та користувацького контенту

Однією з ключових особливостей системи «CNC Helper» є її відкритість до розширення самими користувачами. Оскільки технології обробки матеріалів постійно розвиваються, статична база даних швидко втратила б актуальність. Для вирішення цієї проблеми було розроблено підсистему пропозиції та модерації контенту.

3.4.1 Механіка створення пропозицій (User-Generated Content)

Однією з ключових функцій інформаційної системи «CNC Helper» є можливість децентралізованого розширення бази даних безпосередньо її користувачами в межах краудсорсингової моделі. Оскільки процес інтеграції нового режиму обробки вимагає послідовного введення масиву взаємопов'язаних технологічних параметрів, для програмної реалізації даної логіки було застосовано архітектурний патерн «Сцена» (Wizard Scene), який функціонує за принципом скінченного автомата станів користувацької сесії. На відміну від стандартних лінійних обробників системних команд, зазначена технологія дозволяє серверу застосунку фіксувати й верифікувати поточний крок оператора, причому кожен етап генерації пропозиції супроводжується транзакційним записом проміжних даних у типізований об'єкт сесії. Алгоритм функціонування сцени проєктування нового режиму різання складається з декількох послідовних детермінованих фаз, що забезпечують цілісність збору інформації. На початковому етапі ініціалізації сцени система генерує порожній об'єкт пропозиції та переводить автомат у стан очікування вибору матеріалу заготовки. Наступна фаза передбачає вибір об'єктів ієрархічного каталогу, де за допомогою вбудованих фільтрів оператор послідовно визначає матеріал та інструментальне оснащення. Після цього алгоритм переходить до покрокового введення числових параметрів, перемикаючись у режим обробки вхідних текстових повідомлень, у якому користувач послідовно вводить

частоту обертання шпинделя, швидкість подачі та глибину різання за один прохід (див. рис. 3.12).

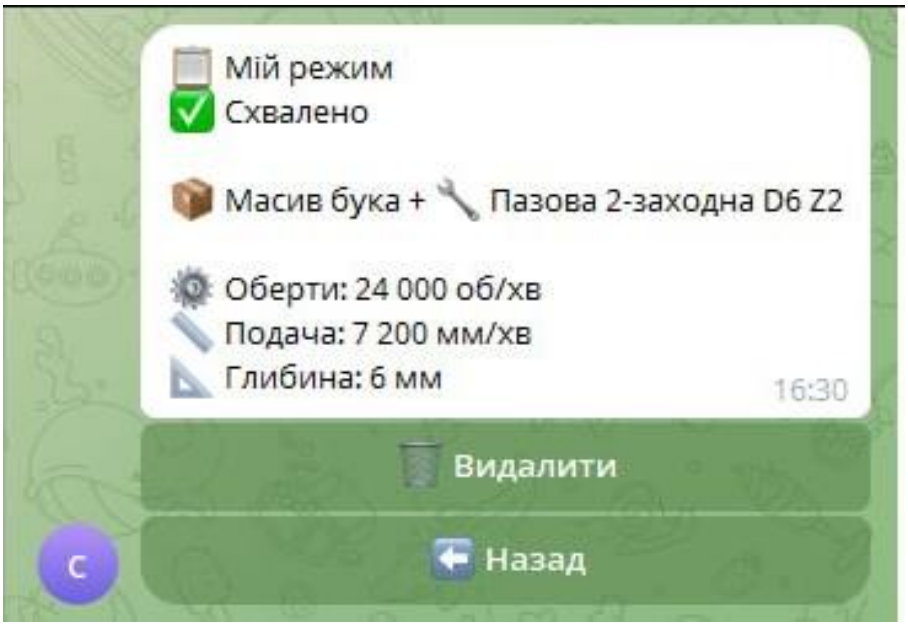


Рисунок *3.12 — Крок введення швидкості подачі у майстрі створення режиму*

Кожна ітерація введення супроводжується роботою проміжного валідатора для відсікання некоректних або фізично неможливих значень. На фінальній стадії агрегування та прев'ю система формує підсумкову картку технологічного режиму, пропонуючи розробнику верифікувати та підтвердити надсилання даних (див. рис. 3.13).



Рисунок *3.13 — Картка перевірки та підтвердження запропонованого технологічного режиму*

Описаний етап підтвердження є критично важливим аспектом проєктування інтерфейсу, оскільки мінімізує ймовірність випадкових розрахункових або синтаксичних помилок перед відправкою даних на сервер. Після успішного проходження етапу підтвердження сформований об'єкт транслюється до сервісу ModerationService, який здійснює персистентний запис у реляційну таблицю CuttingMode з присвоєнням системного статусу PENDING для подальшого адміністрування.

3.4.2 Інтерфейс адміністратора та процес модерації

Для адміністраторів інформаційної системи спроєктовано відокремлений функціональний блок, керування доступом до якого жорстко контролюється на рівні авторизації за допомогою системної ролі ADMIN. При персистенції в реляційній базі даних нових технологічних записів, що маркуються початковим прапором PENDING, обліковий запис адміністратора отримує повноваження на інспектування та опрацювання черги модерації. Спеціалізований інтерфейс адміністративного управління реалізовано у

вигляді уніфікованої інформаційної картки, яка акумулює детальні розрахункові параметри запропонованого режиму різання, ідентифікаційні дані про автора пропозиції для забезпечення внутрішнього моніторингу та телеметрії, а також два інтерактивні елементи керування у вигляді інлайн-кнопок схвалення або відхилення внесених змін.

Активація керуючого елемента «Схвалити» ініціює виконання відповідного методу сервісу бінес-логики, який здійснює транзакційну модифікацію атрибута статусу запису у сховищі даних на APPROVED. Одночасно з цією операцією тригерний механізм зворотного зв'язку через інструментарій Telegram Bot API надсилає автору пропозиції асинхронне сповіщення про успішне проходження верифікації та інтеграцію його технологічного режиму до глобального каталогу, що є елементом стимулювання подальшої краудсорсингової активності інженерного персоналу. У протилежному сценарії, при виборі команди відхилення пропозиції з присвоєнням статусу REJECTED, архітектура системи надає модератору можливість детермінації конкретної причини відмови, яка згодом транслюється автору у вигляді інформаційного повідомлення. Описаний двовекторний підхід забезпечує високий рівень прозорості функціонування інформаційного середовища та гарантує дескриптивну якість і достовірність нормативно-довідкових даних у загальному каталозі. Комплексна логіка взаємодії між користувачем, який виступає ініціатором генерації контенту, реляційною базою даних та адміністратором, що здійснює експертну верифікацію, детально відображена на рисунку 3.14, де наочно продемонстровано послідовність переведення станів об'єкта з фази очікування до безпосередньої публікації.

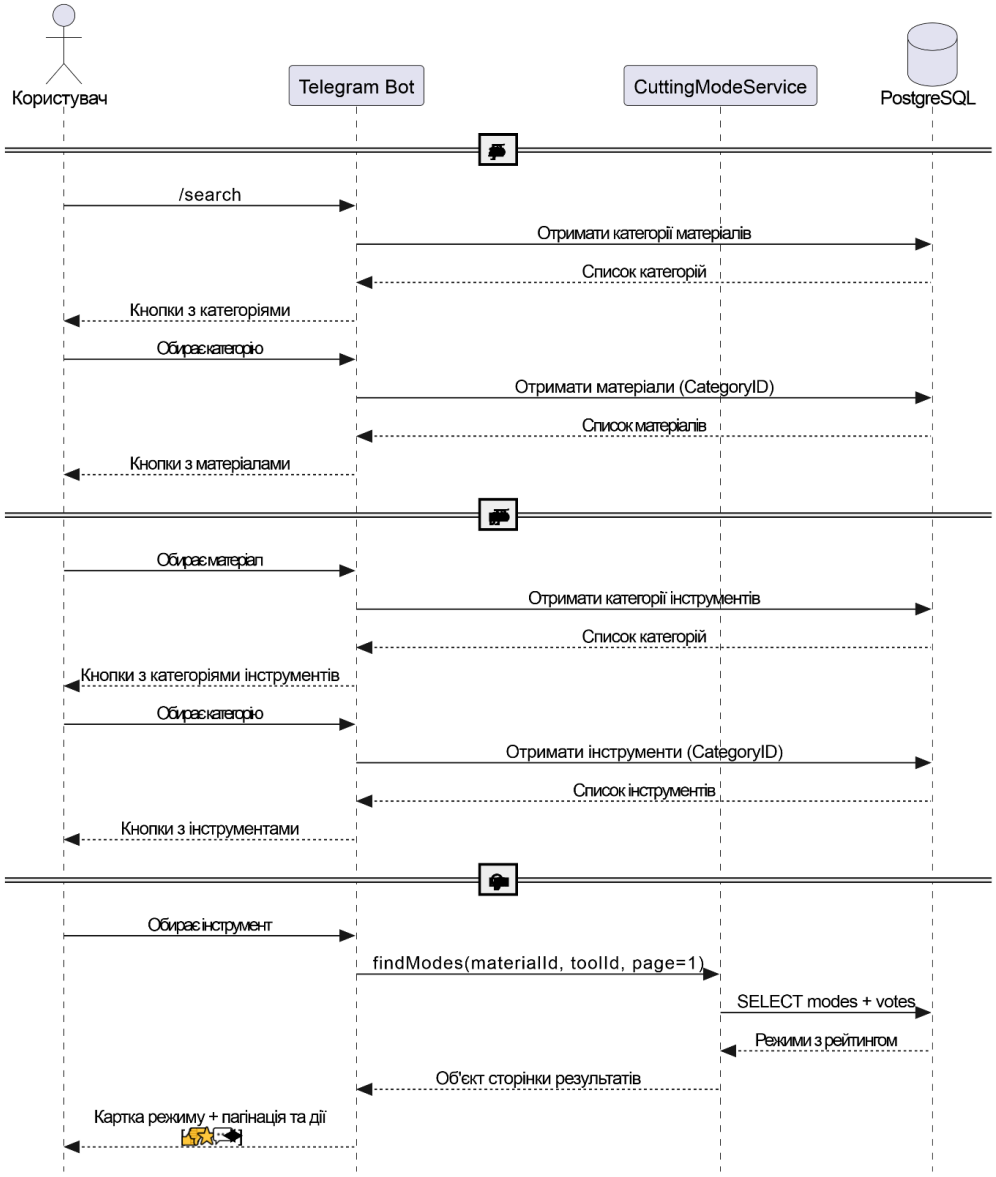


Рисунок *3.14 — Діаграма взаємодії акторів у процесі модерації контенту.*

3.4.3 Персоналізований контроль та обробка винятків

Особливе місце в архітектурі системи посідає персоналізований контроль користувацького контенту. Оператор має доступ до індивідуального розділу «Мої режими», у якому відображаються його власні пропозиції зі статусом верифікації, що забезпечує можливість керування та видалення застарілих даних (див. рис. 3.15).

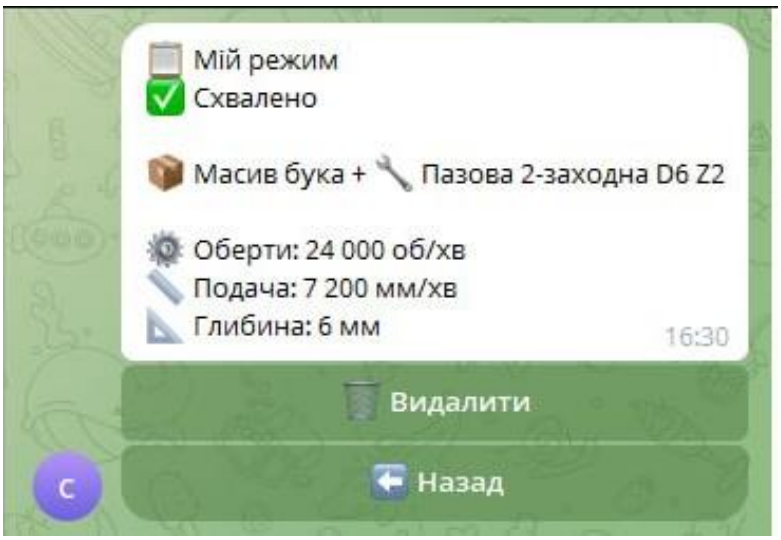


Рисунок *3.15 — Відображення верифікованого режиму в розділі користувача*

У сценаріях, коли для визначеної оператором комбінації конструкційного матеріалу та інструментального оснащення в базі даних PostgreSQL відсутні відповідні записи, система активує захисний алгоритм обробки порожніх значень (див. рис. 3.16).



Рисунок *3.16 — Повідомлення системи про відсутність записів для обраної комбінації*

3.4.4 Управління категоріями та елементами каталогу

Аналогічна логіка застосовується і до розширення самого дерева категорій. Користувачі можуть пропонувати нові типи матеріалів або фрез. Адміністративний модуль дозволяє не лише схвалювати ці пропозиції, а й виконувати пряме редагування каталогу без проходження етапу черги. Це реалізовано через методи PrismaService, які забезпечують атомарність операцій: якщо адміністратор додає новий матеріал, він автоматично стає доступним для всіх користувачів у реальному часі без перезавантаження бота.

3.5 Реалізація персоналізованого розділу «Обране»

Для підвищення ергономічності діяльності операторів верстатів з ЧПК, які переважно функціонують у межах обмеженого масиву типових технологічних завдань, у структурі інформаційної системи реалізовано спеціалізований модуль FavoriteModule. Зазначений функціональний блок дозволяє користувачеві формувати персоналізовану базу швидкого доступу, що складається з верифікованих і найбільш затребуваних режимів механічної обробки. Програмна механіка функціонування даного розділу інтегрує процеси транзакційного додавання, генерації списків та оперативного керування записами. На етапі додавання під кожною інформаційною карткою результатів пошуку відображається інтерактивний керуючий елемент із символічним графічним маркером, активація якого ініціює створення реляційного запису в таблиці Favorite із жорстким зв'язуванням ідентифікаторів user_id та cutting_mode_id.

Процедура перегляду персоналізованих даних реалізована через відповідний інтерфейсний розділ головного меню чат-бота, при виклику якого обчислювальні алгоритми сервера генерують динамічний перелік технологічних режимів, відфільтрованих за ознакою унікального ідентифікатора поточного актора. Процес керування надає оператору повноваження на деструктування зв'язку та вилучення певного режиму з

персонального переліку в будь-який момент експлуатації системи. При цьому архітектурне використання реляційних зв'язків у системі управління базами даних PostgreSQL гарантує збереження та логічну цілісність посилань у таблиці обраного навіть у разі проведення адміністративного оновлення чи модифікації параметрів самого режиму різання, завдяки чому кінцевий користувач асинхронно отримує доступ до найбільш актуальних і верифікованих технологічних даних.

3.6 Валідація даних та обробка виняткових ситуацій

Надійність інформаційної системи, що оперує технологічними параметрами для виробництва, безпосередньо залежить від механізмів обробки помилок та валідації вхідних даних. У системі «CNC Helper» реалізовано багаторівневий захист, що охоплює як рівень користувацького інтерфейсу в Telegram, так і рівень бізнес-логіки на сервері.

3.6.1 Технічна валідація вхідних параметрів

Надійність функціонування інформаційної системи, що оперує нормативно-довідковими технологічними параметрами, безпосередньо зумовлена дескриптивною якістю та коректністю вхідних даних. Оскільки фіксація числових значень частоти обертання шпинделя, швидкості подачі та глибини різання за прохід в інтерфейсі чат-бота здійснюється через стандартне текстове поле, виникає висока ймовірність генерації некоректно сформованих або синтаксично деструктивних запитів. З метою нівелювання зазначеного фактора у структурі модуля ModerationModule реалізовано багаторівневий механізм превентивної валідації, що функціонує за методологічним принципом захисту від помилок (Poka-yoke).

Основним інструментом верифікації вхідного потоку інформації є інтелектуальний парсинг рядкових значень, алгоритм якого інтегрує етапи нормалізації, типізації та фізичного обмеження даних. На початковій стадії

нормалізації рядка програмний комплекс в автоматичному режимі усуває надлишкові пробіли, здійснює уніфікацію десяткових роздільників шляхом заміни коми на крапку для підтримки варіативних мовних локалей, а також відсікає символьні назви одиниць вимірювання у разі їхнього сумісного введення з числовим масивом. Наступна фаза типізації передбачає застосування вбудованих функцій parseFloat() та Number(), причому у сценарії детекції результату перетворення зі значенням NaN (Not a Number) система миттєво перериває виконання поточного алгоритму та транслює користувачеві інформаційне повідомлення про необхідність повторного введення виключно скалярного числового аргументу. Фінальний етап перевірки фізичних меж полягає у зіставленні отриманих величин із допустимими діапазонами предметної області, де, зокрема, частота обертання шпинделя обмежена умовою невід'ємності та верхньою межею технічних можливостей сучасного верстатного обладнання, що регламентує запит на коригування дій оператора при виході значень за встановлені рамки.

Для програмної імплементації зазначених верифікаційних процедур у середовищі NestJS застосовано спеціалізовані об'єкти перенесення даних (Data Transfer Objects — DTO) у синергії з декораторами бібліотеки class-validator. Дане інженерне рішення дозволяє декларативно описувати правила валідації безпосередньо у структурі даних, використовуючи системні анотації на зразок @IsNumber() для гарантування числового типу поля, @Min(0) для превентивної заборони від’ємних значень та @IsNotEmpty() для запобігання відправці порожніх інформаційних форм. Описаний підхід забезпечує високий рівень відмовостійкості програмного комплексу, оскільки навіть у разі некоректного введення параметрів оператором виникнення критичних помилок виконання (Runtime Errors) повністю блокується, застосунок зберігає стабільність функціонування, а користувач отримує чітку інструкцію щодо виправлення помилок, що суттєво оптимізує досвід взаємодії та гарантує чистоту й цілісність інформаційного фонду, що імпортується до реляційної

бази даних PostgreSQL.

3.6.2 Обробка логічних конфліктів та стану бази даних

На рівні взаємодії з реляційною базою даних через інструментарій Prisma ORM реалізовано комплексні програмні механізми захисту від дублювання інформації та виникнення транзакційних конфліктів, що забезпечує високу надійність і цілісність сховища. Зокрема, застосування обмежень унікальності (Unique Constraints) детермінує, що будь-яка спроба повторної реєстрації облікового запису користувача з уже наявним у базі даних ідентифікатором telegram_id автоматично блокується безпосередньо на рівні системи управління базами даних, при цьому спеціалізований сервіс бізнес-логіки перехоплює цю подію та асинхронно повертає існуючий запис замість генерації надлишкової копії. Поряд із цим, контроль цілісності зовнішніх ключів (Foreign Key Integrity) гарантує стабільність системи при роботі з деструктивними операціями: у сценарії, коли оператор здійснює пошук технологічного режиму за ідентифікатором матеріалу, який у цей самий час був видалений або архівований адміністратором, архітектура застосунку забезпечує перехоплення виниклої системної помилки PrismaClientKnownRequestError. Це дозволяє уникнути аварійного завершення потоку виконання, локалізувати збій на рівні сервісу та коректно надіслати користувачеві релевантне інформаційне сповіщення про проведення поточних регламентних робіт чи оновлення структури даного розділу каталогу.

3.6.3 Обробка помилок Telegram API

Взаємодія з Telegram Bot API має свої обмеження (наприклад, ліміти на кількість повідомлень за секунду — Rate Limits). Для стабільної роботи системи використовується вбудований у grammY механізм errorHandler. Він дозволяє централізовано логувати всі помилки мережевого з’єднання або некоректні виклики методів API, не зупиняючи при цьому загальний життєвий цикл застосунку NestJS.

3.7 Автоматизоване наповнення бази даних (Seed-дані)

Для введення розробленого програмного комплексу в експлуатацію, проведення комплексного тестування та демонстрації його функціональних можливостей було спроєктовано й реалізовано механізм автоматизованого наповнення реляційної бази даних первинною нормативно-довідковою інформацією, відомий як процедура сидінгу (Seed). Інтеграція цього інструменту дозволяє повністю нівелювати потребу в ручному покроковому введенні технологічних масивів даних на етапі первинного розгортання системи, а також гарантує асинхронну наявність базового верифікованого набору режимів обробки, необхідного для оцінки відмовостійкості та коректності роботи обчислювальних алгоритмів. Програмна логіка реалізації відповідного скрипта імплементована у структурі файлу prisma/seed.ts і базується на послідовному виклику методів PrismaService для забезпечення транзакційного формування та персистенції повного ієрархічного дерева сутностей предметної області.

Процес генерації первинного інформаційного фонду охоплює три послідовні детерміновані стадії, що починаються зі створення розгалуженої ієрархії конструкційних матеріалів, у межах якої до сховища даних імпортуються базові класифікаційні категорії, зокрема метали, пластики та деревина, а також релевантні їм конкретні матеріали заготовок, такі як алюміній, оргскло або березова фанера, з одночасним присвоєнням їм активованого системного статусу APPROVED. На наступній стадії здійснюється автоматизоване наповнення каталогу різального інструменту, що передбачає реєстрацію переліку типових конфігурацій кінцевих фрез, включаючи пазові, конічні та радіусні моделі, з точним фіксуванням їхніх номінальних діаметрів і супутніх геометричних характеристик. Фінальна фаза полягає у безпосередньому формуванні стійких технологічних зв’язок, де для кожної унікальної пари об'єктів «матеріал + інструмент» генерується щонайменше один реальний режим механічної обробки, параметри якого

екстрапольовані з офіційних нормативних довідників та каталогів провідних виробників інструментального оснащення. Завдяки впровадженню описаного інженерного рішення будь-який розробник, модератор або тестувальник має можливість ініціювати розгортання абсолютно ідентичної, валідної копії інформаційного середовища шляхом виконання стандартизованої консольної команди npx prisma db seed, що забезпечує високий рівень відтворюваності, ізольованості та стандартизації умов розробки, відлагодження та демонстрації програмного продукту.

3.8 Тестування та розгортання системи

Заключним етапом розробки є перевірка працездатності всіх компонентів та підготовка до стабільної експлуатації.

3.8.1 Функціональне тестування

Процес комплексного функціонального тестування інформаційної системи «CNC Helper» охоплював ретельну верифікацію всіх базових користувацьких сценаріїв (Use Cases), попередньо структурованих та описаних у межах теоретичного проєктування другого розділу. У ході проведення випробувань розробленого програмного забезпечення першочергова увага приділялася дескриптивному аналізу відмовостійкості навігаційних алгоритмів, точності обробки пошукових транзакцій та логічній цілісності підсистеми соціальної взаємодії. Верифікація модуля навігації передбачала перевірку коректності та швидкодії динамічних переходів між ієрархічними рівнями інтерактивного меню скінченного автомата, а також стабільності відпрацювання функціональних елементів повернення до попереднього стану сесії. У розрізі тестування пошукового інструментарію здійснювалася строга валідація релевантності вихідних даних, що полягала в оцінці відповідності генерованих сервером режимів різання специфічним фізико-механічним характеристикам обраних оператором конструкційних

матеріалів та геометричних конфігурацій інструменту. Наостанок, особливий акцент було зроблено на аудиті транзакційної логіки модуля голосування, де в умовах симуляції паралельних запитів перевірялася ефективність програмних обмежень, спрямованих на превентивне блокування та унеможливлення повторного нарахування оцінок або дублювання статистичних маркерів від одного й того самого унікального профілю користувача.

3.8.2 Контейнеризація за допомогою Docker

Для забезпечення повної інваріантності від операційного оточення, підвищення загальної стабільності функціонування та суттєвого спрощення процедури розгортання на віддалених серверних потужностях, архітектуру інформаційної системи «CNC Helper» було піддано комплексній контейнеризації. Дане інженерне рішення дозволяє повністю інкапсулювати всі системні залежності застосунку, включаючи середовище виконання Node.js, бінарні системні бібліотеки та інструментарій Prisma ORM, у межах ізольованого програмного образу. Практична архітектура контейнеризації реалізована через розробку оптимізованого Dockerfile, налаштування механізмів оркестрації та конфігурування внутрішньої мережевої взаємодії компонентів.

У розрізі створення розгортального образу для серверної частини на базі NestJS було спроєктовано багатоетапний (multi-stage) Dockerfile. На початковій стадії збирання (build) здійснюється розгортання повного спектра розробницьких залежностей та транскомпіляція вихідного тексту з мови TypeScript у високоефективний JavaScript-код продуктивного типу. На наступній стадії продуктивної експлуатації (production) залучається полегшений базовий образ (node:alpine), у структуру якого імпортуються виключно скомпільовані артефакти та мінімально необхідний набір модулів виконання, що дозволяє зменшити підсумковий геометричний розмір образу в кілька разів, безпосередньо прискорюючи процес його трансляції мережею та розгортання. Оркестрація багаторівневої структури системи, що поєднує

рівень збереження даних та рівень бізнес-логіки, реалізована за допомогою інструменту Docker Compose. У межах конфігураційного файла docker-compose.yml детерміновано функціонування двох базових сервісів: db, який функціонує на основі офіційного образу СКБД PostgreSQL і використовує механізм монтування томов (volumes) для забезпечення персистентності та збереження даних після зупинки контейнера на фізичному носії сервера, а також api (безпосередньо застосунок NestJS), запуск якого логічно обмежений директивою depends_on, що жорстко регламентує правильну хронологічну послідовність ініціалізації системних компонентів.

Внутрішня мережева взаємодія елементів платформи ізольована у межах створеної віртуальної мережі типу bridge, завдяки чому сервер застосунку отримує можливість здійснювати транзакційні запити до СКБД за допомогою внутрішнього мережевого імені сервісу db замість використання статичних IP-адрес. Описане рішення суттєво підвищує гнучкість адміністрування та рівень інформаційної безпеки, оскільки усуває необхідність прокидання системного порту бази даних у зовнішній контур сервера. Додатково у процесі розгортання контейнера автоматизовано алгоритм виконання міграцій: при кожній ініціалізації середовища застосунок аналізує поточний стан реляційної схеми і за наявності девіацій асинхронно виконує команду prisma migrate deploy, що гарантує автоматичну актуалізацію структури таблиць при кожній ітерації оновлення версії програмного продукту та повністю нівелює ризики виникнення помилок несумісності даних на рівні сховища.

3.9 Висновки до розділу 3

Сукупний аналіз результатів проєктування та розробки дозволяє констатувати, що всі поставлені науково-технічні та прикладні завдання були виконані у повному обсязі на кожному етапі дослідження. На стадії проєктування проведено глибокий аналіз предметної області та специфікації

функціональних вимог, що забезпечило формування оптимальної, слабко пов'язаної архітектури системи, а вибір сучасного інженерного стеку технологій у складі програмного середовища Node.js, фреймворку NestJS та системи управління базами даних PostgreSQL став надійним фундаментом для створення високопродуктивного й масштабованого рішення. Розроблена концептуальна ER-діаграма бази даних врахувала всі необхідні реляційні зв’язки та обмеження цілісності, необхідні для ефективної реалізації складного ієрархічного пошуку, краудсорсингових механізмів та підсистеми соціальної взаємодії користувачів.

На етапі безпосередньої програмної реалізації було успішно створено повнофункціональний робочий прототип інформаційної системи, який чітко відповідає всім заявленим експлуатаційним критеріям та технічним умовам. Особливу увагу в процесі розробки приділено UX-складовій інтерфейсу чат-бота месенджера Telegram, у межах якої імплементовано детерміновані сценарії взаємодії на основі скінченних автоматів, інтегровано систему динамічної пагінації повідомлень для оптимізації екранного простору та розроблено персоналізовані модулі збереження даних. Застосування інструментів автоматизації розгортання, зокрема скрипта первинного наповнення Prisma Seed, дозволило асинхронно імпортувати до сховища еталонні масиви нормативно-довідкових технологічних даних для негайного введення платформи в експлуатацію та проведення натурних випробувань.

Таким чином, інформаційна система «CNC Helper» — це цілісний і відмовостійкий програмний продукт, що поєднує потужність реляційних баз даних із гнучкістю мобільного кросплатформного інтерфейсу. Модульна архітектура сервера має високий потенціал для масштабування, що дозволить у майбутньому легко інтегрувати нові модулі: від аналітичного розрахунку режимів різання до обміну файлами керуючих програм через хмарні сховища

ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальне науково-практичне завдання, що полягає в проєктуванні та програмній реалізації інформаційної системи у вигляді кросплатформного чат-бота для автоматизації процесів пошуку, підбору та диференційованого управління режимами різання на верстатах з числовим програмним керуванням (ЧПК). Завдяки залученню сучасного стеку вебтехнологій створено мобільне інженерне рішення, яке суттєво оптимізує оперативний доступ операторів до нормативно-довідкової технологічної інформації безпосередньо в цехових умовах промислового виробництва.

У ході виконання дослідження проведено системний аналіз предметної області та існуючих програмних рішень, за результатами якого встановлено, що традиційні паперові довідники та стаціонарні CAM-системи не забезпечують належного рівня мобільності та можливості оперативного обміну практичним досвідом між фахівцями, що обґрунтувало доцільність розробки сервісу на базі месенджера Telegram, яка усуває потребу в проєктуванні окремих клієнтських застосунків під різні операційні системи. На основі формалізованих вимог для ролей оператора та адміністратора спроєктовано архітектуру системи, центральним елементом якої є концептуальна реляційна модель бази даних, що складається з 9 взаємопов'язаних сутностей. Інтеграція системи управління базами даних PostgreSQL дозволила ефективно реалізувати складні ієрархічні зв'язки між конструкційними матеріалами, геометрією інструментального оснащення та параметрами обробки із забезпеченням суворої цілісності даних на рівні обмежень зовнішніх ключів та унікальних індексів сховища. Програмна реалізація серверної частини виконана на базі фреймворку NestJS та спеціалізованої бібліотеки grammY, де безпечний і строго типізований доступ до таблиць забезпечено інструментарієм Prisma ORM, а процес покрокової

селекції параметрів організовано на основі математичної моделі скінченного автомата з передачею контексту станів через оптимізовані рядки callback_data.

Для верифікації інформаційного фонду впроваджено підсистему соціальної взаємодії та багаторівневого контролю якості контенту, яка включає алгоритми оцінювання режимів за логікою транзакційного оновлення (UPSERT), що унеможливлює штучне дублювання голосів від одного профілю, а також модуль модерації, який ізолює користувацькі пропозиції у чергу очікування зі статусом PENDING до моменту їхньої експертної перевірки адміністратором. Високий рівень відмовостійкості та безпеки програмного комплексу досягнуто за рахунок інтеграції DTO-валідації вхідних рядків і перехоплення виняткових ситуацій Telegram API, а процеси розгортання та незалежності від операційного середовища оптимізовано шляхом повної контейнеризації застосунку за допомогою Docker. Додатково у роботі проаналізовано умови праці розробника програмного забезпечення, виконано інженерні розрахунки штучного освітлення робочої зони, визначено нормативні вимоги щодо електробезпеки та пожежної профілактики, а також детально опрацьовано алгоритми дій в умовах надзвичайних ситуацій воєнного стану, орієнтовані на збереження логічної цілісності баз даних, резервування інформаційних ресурсів та забезпечення фізичної безпеки обслуговуючого персоналу. Складена сукупність отриманих результатів свідчить, що розроблений програмний продукт «CNC Helper» повністю відповідає критеріям технічного завдання, володіє високим потенціалом для практичного впровадження на машинобудівних підприємствах та має гнучку архітектуру для подальшої модернізації, зокрема в напрямку інтеграції модулів автоматизованого аналітичного розрахунку технологічних параметрів за математичними формулами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Грановський Г. І., Грановський В. Г. Різання металів : підручник. Київ : Техніка, 1985. 304 с.
2. Родін П. Р. Основи проєктування різальних інструментів. Київ : Вища школа, 1990. 424 с.
3. Внуков Ю. М. Теорія різання : підручник. Запоріжжя : ЗНТУ, 2011. 340 с.
4. Пижов І. М., Клименко Г. П. Теорія різання матеріалів та різальний інструмент : навч. посібник. Харків : НТУ «ХПІ», 2018. 308 с.
5. Третьяков В. М. CAM системи в машинобудуванні : монографія. Київ : НТУУ «КПІ», 2019. 256 с.
6. Павлов А. О. Розробка чат ботів для бізнесу : практ. посібник. Київ : Комп'ютерпрес, 2023. 192 с.
7. Сітнік В. Ф. Інформаційні системи в менеджменті : підручник. Київ : КНЕУ, 2020. 452 с.
8. Гамма Е., Хелм Р., Джонсон Р., Вліссідес Дж. Патерни проєктування. Київ : Символ Плюс, 2021. 368 с.
9. Фаулер М. Шаблони корпоративних застосунків. Київ : Символ Плюс, 2020. 544 с.
10. Telegram Bot API Documentation. URL: https://core.telegram.org/bots/api (дата звернення: 15.04.2026).
11. grammY Documentation. URL: https://grammy.dev/ (дата звернення: 15.04.2026).
12. NestJS Documentation. URL: https://docs.nestjs.com/ (дата звернення: 15.04.2026).
13. Prisma ORM Documentation. URL: https://www.prisma.io/docs (дата звернення: 15.04.2026).
14. PostgreSQL Documentation. URL: https://www.postgresql.org/docs/ (дата звернення: 15.04.2026).

15. TypeScript Handbook. URL: https://www.typescriptlang.org/docs/ (дата звернення: 15.04.2026).
16. Михайличенко Я.А., бакалавр, Міхайлуца О.М., доцент — науковий керівник. РОЗРОБКА ВЕБПЛАТФОРМИ ДЛЯ ОБМІНУ ДОСВІДОМ МІЖ ОПЕРАТОРАМИ ЧПУМолода наука- 2026 : зб. наук. праць студентів, аспірантів і молодих вчених. Запоріжжя : ЗНУ, 2026. Т. 5. С. 413–414.
17. Михайличенко Я.А., Міхайлуца О.М. Програмний комплекс для інтерактивного обміну виробничими даними операторів ЧПУ на базі архітектури NestJS. Інформаційні технології: теорія і практика: Матеріали III (IX) Міжнародної інтернет-конференції здобувачів вищої освіти і молодих учених (Харків-Запоріжжя-Дніпро, 25–27 березня 2026 р.).. стор. 577-580.

Декларація

академічної доброчесності

здобувача ступеня вищої освіти ЗНУ

Я, Михайличенко Ярослав Андрійович, студент 4 курсу, форми навчання денної, Інженерного навчально-наукового інституту ім. Ю.М. Потебні ЗНУ, спеціальність 121 Інженерія програмного забезпечення освітньої програми «Програмне забезпечення систем», адреса електронної пошти arostar197541@gmail.com, — підтверджую, що написана мною кваліфікаційна робота на тему «Розробка веб-платформи для професійного обміну параметрами обробки та оптимізації процесів на CNC-верстатах» відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст.42 Закону України «Про освіту», зі змістом яких ознайомлений.

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;

-згоден на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою інтернет-системи, а також на архівування моєї роботи в базі даних цієї системи.

Дата 07.06.2026 _____ Михайличенко Ярослав Андрійович
(студент)

Дата 07.06.2026 _____ Міхайлуца Олена Миколаївна
(керівник)