

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджментуКафедра ІнформатикиРівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУздобувачеві Долгому Євгенію Олександровичу
(прізвище, ім'я, по батькові)1. Тема роботи Розроблення інтелектуального застосунку розрахунку дози інсуліну для хворих на цукровий діабет І типу з функціями прогнозування та формування рекомендацій

затверджена наказом університету від 26 травня 2026 року № 435Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 29 червня 2026 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Аналіз сучасних підходів прогнозування та управління глікемією.

2. Розробка та реалізація математичної моделі розрахунку доз інсуліну та прогнозування глікемії.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми прогнозування глікемії, постановка задачі, тестові дані.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

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

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	13.04.2026	
2	Аналіз завдання, підбір літератури	05.05.26-10.05.26	
3	Аналіз літератури з проблеми, що вирішується	11.05.26-13.05.26	
4	Аналіз існуючих методів прогнозування	14.05.26-15.05.26	
5	Формування кроків вирішення проблеми	16.05.26-17.05.26	
6	Програмна реалізація	18.05.26-28.05.26	
7	Аналіз отриманих результатів	29.05.26-01.06.26	
8	Оформлення пояснювальної записки	02.06.26-09.06.26	
9	Перевірка на нормоконтроль	10.06.26-19.06.26	
10	Перевірка на плагіат	11.06.26-30.06.26	
11	Рецензування	20.06.26-30.06.26	
12	Підготовка презентації та доповіді	20.06.26-30.06.26	
13	Занесення роботи в електронний архів	30.06.26-09.07.26	
14	Попередній захист кваліфікаційної роботи	30.06.26-09.07.26	

Дата видачі завдання 13 квітня 2026 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Любченко В. А.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 92 с., 9 табл., 25 рис., 3 дод., 22 джерела.

ПРОГНОЗУВАННЯ ГЛІКЕМІЇ, ЦУКРОВИЙ ДІАБЕТ І ТИПУ, DOCKER, GEMINI API, IONIC, KOTLIN, SPRING BOOT.

Об'єктом роботи є розроблення застосунку для розрахунку доз інсуліну, прогнозування глікемії та формування рекомендацій для хворих на цукровий діабет І типу.

Метою роботи є розроблення застосунку для розрахунку болюсної дози інсуліну на основі рівня глюкози, спожитих вуглеводів та параметрів пацієнта з використанням машинного навчання для прогнозування й рекомендацій.

Під час роботи використано методи штучного інтелекту та машинного навчання, LLM для генерації рекомендацій, клієнт-серверний підхід.

Практична цінність застосунку полягає в автоматизації розрахунку доз інсуліну та зменшенні ризику помилок.

Розроблено застосунок для збору й аналізу даних про рівень глюкози, прогнозування стану пацієнта та формування адаптивних рекомендацій.

Область застосування: цифрова медицина, системи підтримки прийняття рішень, моніторинг здоров'я.

DOCKER, GEMINI API, GLYCEMIA FORECASTING, IONIC, KOTLIN, SPRING BOOT, TYPE 1 DIABETES.

The object of the work is the development of an application for insulin dose calculation, glycemic prediction, and recommendation generation for patients with type I diabetes mellitus.

The purpose of the work is to develop an application for bolus insulin dose calculation based on glucose levels, consumed carbohydrates, and patient parameters using machine learning methods for prediction and recommendations.

During the work, artificial intelligence and machine learning methods, LLM for recommendation generation, client-server architecture were used.

The practical value of the application lies in automating insulin dose calculation and reducing calculation errors.

An application has been developed for glucose data collection and analysis, patient state prediction, and adaptive recommendation generation.

Scope of application: digital medicine, decision support systems, and health monitoring.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	7
Вступ.....	8
1 Аналіз предметної області та методів управління глікемією.....	9
1.1 Особливості інсулінотерапії при діабеті I типу.....	9
1.2 Огляд існуючих систем моніторингу та розрахунку доз.....	11
1.3 Методи штучного інтелекту в задачах ендокринології	14
1.3.1 Використання регресійних моделей для аналізу чутливості до інсуліну	14
1.3.2 Великі мовні моделі як інструмент формування персоналізованих рекомендацій.....	16
1.4 Порівняльний аналіз алгоритмів прогнозування рівня глюкози ...	19
1.5 Постановка задачі	21
2 Аналіз та розробка математичної моделі.....	23
2.1 Математичне моделювання розрахунку болюсної дози.....	23
2.1.1 Модель врахування вуглеводного коефіцієнта та фактора чутливості	24
2.1.2 Алгоритм корекції дози на основі прогнозованої фізичної активності	27
2.2 Алгоритми прогнозування глікемічних станів	32
2.3 Методологія інтеграції LLM для формування рекомендацій	33
2.4 Модель безпеки розрахованих даних	38
2.5 Перевірка коректності роботи моделі.....	41
3 Комп'ютерна модель інтелектуального застосунку	45
3.1 Обґрунтування вибору середовища програмної реалізації	45
3.2 Програмна реалізація.....	49
3.2.1 Архітектура системи.....	49
3.2.2 Розробка серверного застосунку	51
3.2.3 Розробка модулів обчислень та штучного інтелекту	57

	6
3.2.4 Розробка клієнтського застосунку	65
3.3 Інструкція користувача	67
Висновки	81
Перелік джерел посилання	83
Додаток А Тестова вибірка детермінованого алгоритму розрахунку болюсу	86
Додаток Б Тестова вибірка оцінки вмісту вуглеводів мовною моделлю	89
Додаток В Тестова вибірка клінічної безпечності рекомендацій	91

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

МОЗ – Міністерство Охорони Здоров'я

ACSM – American College of Sports Medicine

ADA – American Diabetes Association (американська діабетична асоціація)

ARIMA – AutoRegressive Integrated Moving Average

HbA1c – Hemoglobin A1C, glycated hemoglobin (глікований гемоглобін)

HTTP – Hypertext Transfer Protocol

ICR – Insulin-to-Carbohydrate Ratio

ISF – Insulin Sensitivity Factor

JNA – Java Native Access

JPA – Java Persistence API

JVM – Java Virtual Machine

JWT – JSON Web Token

LLM – Large Language Model (велика мовна модель)

LSTM – Long Short-Term Memory

MAE – Mean Absolute Error

RAG – Retrieval-Augmented Generation

TIR – Time In Range

ВСТУП

Цукровий діабет I типу є хронічним ендокринним захворюванням, яке характеризується порушенням вироблення інсуліну та потребує постійного контролю рівня глюкози в крові. Для підтримання стабільного стану пацієнтам необхідно регулярно здійснювати розрахунок доз інсуліну з урахуванням рівня глікемії, кількості спожитих вуглеводів, фізичної активності та індивідуальних особливостей організму.

Сучасні цифрові технології у сфері медицини активно застосовуються для моніторингу стану здоров'я, аналізу медичних показників та підтримки прийняття рішень. Одним із перспективних напрямків є створення інтелектуальних застосунків, які здатні автоматизувати розрахунки, прогнозувати зміни стану пацієнта та формувати персоналізовані рекомендації. Для цього використовуються методи штучного інтелекту, машинного навчання та обробки природної мови, що дозволяють аналізувати великі обсяги даних і підвищувати точність прогнозування.

Одним із важливих завдань лікування цукрового діабету I типу є правильний розрахунок дози інсуліну, оскільки помилки можуть призводити до гіпоглікемічних або гіперглікемічних станів. Використання інтелектуальних систем дає можливість мінімізувати ризики людської помилки та покращити ефективність контролю захворювання.

Актуальність роботи полягає у необхідності розроблення інтелектуального застосунку для розрахунку дози інсуліну хворим на цукровий діабет I типу з функціями прогнозування рівня глюкози та формування рекомендацій на основі аналізу індивідуальних показників пацієнта.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА МЕТОДІВ УПРАВЛІННЯ ГЛІКЕМІЄЮ

1.1 Особливості інсулінотерапії при діабеті I типу

Цукровий діабет I типу – це захворювання, пов'язане з імуніопосередкованим руйнуванням β -клітин підшлункової залози, що виробляють інсулін, що призводить до дефіциту інсуліну. Цукровий діабет I типу менш поширений у дорослих, ніж у молодих людей, і його слід відрізняти від діабету II типу, який більш поширений у літніх людей. Люди з цукровим діабетом I типу потребують довічної заміни екзогенного інсуліну. Без інсуліну у пацієнтів може розвинутися важка гіперглікемія і, зрештою, діабетичний кетоацидоз, який може бути небезпечним для життя. Додатковою складністю інсулінотерапії є моніторинг рівня глюкози, спеціальна освіта, дієта та спосіб життя.

Природний перебіг і розвиток діабету I типу проходить у 3 стадії.

Стадія 1, доклінічна стадія, характеризується початком аутоімунного руйнування β -клітин та інсуліту, спричиненого імуніопосередкованим руйнуванням. Ця стадія протікає безсимптомно і характеризується нормальним рівнем глюкози натще, нормальною толерантністю до глюкози та наявністю щонайменше 2 панкреатичних аутоантитіл.

На стадії 2 вже виникає значна дисфункція β -клітин, що призводить до дисглікемії. Діагностичні критерії включають наявність панкреатичних аутоантитіл із порушенням рівня глюкози натще (рівень глюкози натще 5,6-6,9 ммоль/л), порушення толерантності до глюкози (2 години після навантаження глюкозою 75 г глюкози 7,8-11,1 ммоль/л) або рівень глікованого гемоглобіну (HbA1c) від 5,7% до 6,4%. Люди залишаються безсимптомними [1].

Стадія 3 характеризується клінічним початком захворювання, коли в осіб спостерігається симптоматична гіперглікемія. Діагностичні критерії

включають діабет, що визначається гіперглікемією (випадковий рівень глюкози $\geq 11,1$ ммоль/л) із клінічними симптомами, глюкозою натще принаймні 7,0 ммоль/л, рівнем глюкози в крові принаймні 11,1 ммоль/л через 2 години після прийому 75 г глюкози під час перорального тесту на толерантність до глюкози або HbA1c більше або дорівнює 6,5%. У осіб із класичним нововиявленим діабетом I типу зазвичай спостерігаються симптоми полідипсії, поліурії, поліфагії, ненавмисної втрати ваги, втоми та слабкості. Загрозливі для життя ускладнення може розвинути, якщо не лікувати негайно.

Основним методом лікування на клінічній стадії є інтенсивна інсулінотерапія, яка передбачає поєднання інсуліну тривалої дії (базального) та короткої дії (болюсного). Сучасна інсулінотерапія базується на використанні аналогів інсуліну, які отримують за допомогою синтетичних модифікацій природнього інсуліну. Такі аналоги інсулінів короткої дії потрапляють у кровотік протягом декількох хвилин після введення, тому вони вводяться за декілька хвилин до прийому їжі. Час до початку дії інсуліну може залежати від препарату, особистих параметрів пацієнту, таких як вага, вік та поточного значення HbA1c [2]. Поєднання базального та болюсного інсуліну дозволяє максимально наблизити терапію до природньої роботи підшлункової залози. Базальний підтримує стабільний фоновий рівень глюкози, тоді як болюсний корегує на конкретний прийом їжі чи відхилення від норми.

Інсулінотерапія сама по собі несе суттєвий ризик для здоров'я, що перетворює кожен розрахунок дози інсуліну не просто на математичну задачу, а на рішення з прямими наслідками для здоров'я та життя пацієнта. Гіпоглікемія порушує фізіологічні та поведінкові захисні механізми, що збільшує ризик повторних гіпоглікемічних епізодів і їхніх ускладнень. Гіпоглікемія спричиняється передозуванням інсуліну в поєднанні з дефектною контррегуляторною відповіддю та, якщо її не лікувати, може призвести до судом, втрати свідомості або смерті [3]. Таким чином, пацієнт із

діабетом I типу постійно балансує між двома протилежними небезпеками гіперглікемією при недостатній дозі та гіпоглікемією при надмірній.

Розрахунок болюсної дози є критично складним процесом, оскільки потребує врахування не лише поточної глікемії та кількості вуглеводів, а й індивідуальних коефіцієнтів чутливості, рівня активного інсуліну в крові та майбутньої фізичної активності. Це обумовлює необхідність автоматизації таких розрахунків за допомогою інтелектуальних систем. Також автоматизація медичних процесів на основі методів аналізу даних дозволяє суттєво скоротити час діагностики та мінімізувати вплив людського фактору на результати [4].

1.2 Огляд існуючих систем моніторингу та розрахунку доз

Інтенсивна інсулінотерапія для лікування діабету I типу була встановлена як стандарт лікування на основі результатів дослідження контролю діабету та ускладнень. Однак інтенсивне лікування інсуліном, може викликати гіпоглікемію, яка може бути гостро небезпечною для життя та призвести до судом, коми або летального наслідку. Американська діабетична асоціація (ADA) рекомендує рівень глікованого гемоглобіну (HbA1c) менше 7% у дорослих.

Щоб досягти цих рекомендованих глікемічних цілей, пацієнти повинні контролювати рівень глюкози в крові кілька разів на день, ретельно оцінювати споживання вуглеводів. Системи моніторингу рівня глюкози бувають 3 типів: традиційні глюкометри, системи безперервного моніторингу в реальному часі та системи флеш-моніторингу.

Традиційні глюкометри вимірюють рівень глюкози в капілярній крові в конкретний момент часу. Зазвичай це переносний пристрій, який вимірює і зберігає дані за допомогою тест-смужок. Для використання таких

глюкометрів пацієнту потрібно використовувати ручку-проколювач, для отримання невеликої кількості крові.

Системи безперервного моніторингу – це технологія, яка забезпечує безперервні вимірювання рівня глюкози в режимі реального часу з опціями сповіщень про гіперглікемію, гіпоглікемію або прогнозований рівень глюкози поза цільовими діапазонами.

Системи флеш-моніторингу – це технологія моніторингу, в якій дані зберігаються в переносному датчику та отримані шляхом сканування датчика за допомогою спеціального приймача або смартфона.

Традиційні глюкометри офіційно вважаються застарілою технологією і зазвичай використовуються серед діабетиків II типу, але технологія все ще найбільш поширена серед діабетиків I типу в країнах з низьким економічним рівнем. В розвинених країнах здебільшого використовуються системи безперервного моніторингу.

Серед найбільш поширених представників систем безперервного моніторингу є Dextom G7 та FreeStyle Libre 3. Системи автоматично передають показники рівня глюкози на смартфон кожну хвилину та надсилають сповіщення про критичні стани. Серед розвинених країн ці представники є найпоширенішими системами моніторингу.

Поряд із показником HbA1c, сучасні системи моніторингу орієнтуються на метрику «час у діапазоні» (TIR, Time in Range). Це відсоток часу, протягом якого рівень глюкози знаходиться в межах 3,9–10,0 ммоль/л. Цільовим значенням для більшості дорослих із діабетом I та II типу є TIR не менше 70% доби, що відповідає приблизно 17 годинам на добу в цільовому діапазоні. TIR є сильним показником ризику довгострокових ускладнень [5].

Системи розрахунку доз (калькулятори болюса) призначені для визначення точної кількості інсуліну, необхідної для компенсації спожитих вуглеводів та корекції поточного рівня глюкози. Такі системи зазвичай інтегровані в інсулінові помпи, які в свою чергу підключені до систем моніторингу і автоматично регулюють рівень глюкози. Серед

найпоширеніших реалізацій таких систем є Bolus Wizard, який зазвичай вбудований в помпи Medtronic MiniMed. Ця реалізація розраховує рекомендовану дозу на основі поточного рівня глюкози, фактору чутливості та активного інсуліну від попередніх ін'єкцій [5].

Найпростішим способом розрахунку є розрахунок за формулою. Цей спосіб використовується лікарями ендокринологами та більшістю систем для швидкого розрахунку з відносно високою точністю результату. Проте більшість стандартних калькуляторів, вбудованих в інсулінові помпи або мобільні застосунки, використовують детерміновану математичну модель. Але така формула не враховує зовнішні фактори, такі як фізична активність, якість сну, тип продуктів.

$$BC = \frac{C}{ICR} + \frac{BG - T}{ISF} - IOB, \quad (1.1)$$

де BC – кількість одиниць інсуліну швидкої дії, яку пацієнт має ввести;

C – сумарна вага вуглеводів у грамах;

ICR – показує, скільки грамів вуглеводів «компенсує» 1 одиниця інсуліну;

BG – поточний рівень глюкози в крові;

T – рівень цукру, якого пацієнт прагне досягти;

ISF – кількісне зниження глюкози після введення 1 одиниці інсуліну;

IOB – кількість інсуліну від попередніх ін'єкцій, який ще продовжує діяти.

Незважаючи на наявність інструментів для розрахунку болюса, до 64% пацієнтів некоректно розраховують дози інсуліну – переважно через недооцінку вмісту вуглеводів у їжі, що призводить до гіперглікемії та підвищеного ризику гіпоглікемії внаслідок завищених корекційних доз [6].

Сучасні системи розрахунку доз інсуліну еволюціонували від простих калькуляторів до складних цифрових радників, які намагаються враховувати динамічні змінні життєдіяльності пацієнта. Такі рішення дозволяють коригувати дозу залежно від глікемічного індексу продуктів, розрізняючи

«швидкі» та «повільні» вуглеводи. Проте головним обмеженням цих систем залишається необхідність ручного введення великої кількості параметрів та їхня нездатність самостійно адаптуватися до змінних факторів навколишнього середовища без втручання користувача.

Найбільш прогресивним етапом розвитку даної галузі є інтелектуальні системи на базі штучного інтелекту. Завдяки впровадженню алгоритмів машинного навчання система набуває здатності до самоадаптації, виявляючи приховані закономірності у чутливості пацієнта до інсуліну залежно від часу доби чи дня тижня, тоді як використання великих мовних моделей дозволяє автоматично трансформувати текстові описи їжі у конкретні розрахункові дані. Такий підхід забезпечує предиктивну підтримку, прогножуючи ризики виникнення гіпоглікемії за кілька годин до події та надаючи превентивні рекомендації.

Таким чином, жодна з розглянутих існуючих систем не вирішує задачу комплексно – детерміновані калькулятори розраховують дозу за фіксованими коефіцієнтами без урахування індивідуальної адаптації, а існуючі застосунки не забезпечують контекстних пояснень та персоналізованих рекомендацій. Це обумовлює необхідність розроблення застосунку, який поєднає адаптивний розрахунок болюсної дози, прогнозування глікемічних станів та інтелектуальні рекомендації на основі індивідуальних даних пацієнта.

1.3 Методи штучного інтелекту в задачах ендокринології

1.3.1 Використання регресійних моделей для аналізу чутливості до інсуліну

Для забезпечення інтелектуальних функцій підтримки прийняття рішень у системі управління глікемією застосовуються методи статистичного прогнозування. Основними інструментами для адаптації алгоритмів до особливостей є регресійний аналіз.

Регресійні моделі в роботі використовуються для встановлення кількісних закономірностей між вхідними чинниками та результуючим показником рівня глюкози в крові. Вхідними даними можуть бути кількість спожитих вуглеводів, дози інсуліну, фізична активність. При цьому якість результатів аналізу медичних даних суттєво залежить від коректності попередньої обробки вхідної інформації – некоректний вибір методу може призвести до хибних результатів. Тому коректна підготовка та попередня обробка таких даних є необхідною умовою отримання достовірних результатів моделювання [7, 8].

Застосування методу найменших квадратів до історичного масиву даних пацієнта дозволяє уточнювати значення вуглеводного коефіцієнта та індекс інсулін резистентності. Це перетворює статичну медичну формулу на адаптивну модель.

Також можливе короткострокове прогнозування. Методи аналізу часових рядів широко застосовуються в медичних дослідженнях для виявлення прихованих закономірностей у динаміці біологічних показників, що дозволяє діагностувати зміни стану здоров'я на ранніх стадіях [9]. Показники рівня глюкози розглядаються як дискретні часові ряди – послідовності значень, зафіксованих через рівні проміжки часу. Основним завданням цього аналізу є екстраполяція послідовності для завчасного попередження про критичні стани.

Використання аналізу часових рядів дозволяє системі реалізувати функцію прогнозування не лише на основі поточного значення глікемії, а й з урахуванням вектору змін. Такий підхід мінімізує ризики виникнення критичних станів та підвищує загальну безпеку терапії.

Серед регресійних підходів найширше застосування у задачах прогнозування глікемії отримала множинна лінійна регресія. Дослідження показують, що модель на основі лінійної регресії може перевершувати сучасні методи глибокого навчання при прогнозуванні рівня глюкози на горизонті 60 хвилин, зберігаючи при цьому значно меншу обчислювальну

складність [10]. Метод найменших квадратів обирається як основний інструмент побудови таких моделей через його математичну прозорість та стійкість до перенавчання на обмежених вибірках індивідуальних даних пацієнта, що є типовою умовою в персоналізованих медичних застосунках.

Проте динаміка глюкози не завжди підпорядковується лінійним залежностям. Наприклад, після прийому їжі глюкоза спочатку різко зростає, потім знижується під дією інсуліну – це крива, а не пряма. Для таких випадків застосовується поліноміальна регресія, яка на відміну від лінійної, може описувати подібні криволінійні закономірності.

Важливою перевагою поліноміальної регресії є те, що вона залишається математично прозорою. Результат можна пояснити і перевірити, що критично важливо в медичних застосунках. На відміну від нейромережових моделей, які дають точну відповідь але без пояснення чому саме таку, регресійні моделі дозволяють лікарю або розробнику зрозуміти логіку кожного розрахунку. При цьому дослідження показують, що індивідуалізовані лінійні та поліноміальні моделі часто не поступаються нелінійним методам у точності прогнозування глікемії.

Таким чином, для реалізації адаптивного розрахунку коефіцієнтів ICR та ISF у даній роботі обрано підхід на основі лінійної та поліноміальної регресії як метод, що забезпечує достатню точність на обмежених індивідуальних даних пацієнта, зберігає математичну прозорість результату та не потребує значних обчислювальних ресурсів для роботи на мобільному пристрої.

1.3.2 Великі мовні моделі як інструмент формування персоналізованих рекомендацій

Традиційні системи управління обмежені наданням сухих кількісних показників, що може ускладнювати їхню інтерпретацію пацієнтом у

реальних життєвих ситуаціях. Інтеграція великих мовних моделей (Large Language Models, LLM) у структуру інтелектуального застосунку дозволяє перейти від простих розрахунків до формування змістовних персоналізованих рекомендацій природною мовою.

Досвід застосування великих мовних моделей у прикладних системах свідчить про їхню здатність обробляти різноманітні вхідні дані та генерувати структурований контент природною мовою. Зокрема, дослідження показують, що LLM успішно справляються з задачами мультимодального аналізу, де вхідними даними є одночасно текст, аудіо та візуальний контент, а результатом є структурована відповідь, адаптована до конкретного контексту користувача. При цьому якість генерованого контенту оцінюється експертами в середньому на рівні 4,3 з 5 балів, що підтверджує практичну придатність таких систем для реального використання [11]. Важливою особливістю є здатність моделей адаптуватися до контексту – зокрема, скорочувати або розширювати відповідь залежно від обсягу вхідних даних, що є критично важливим для медичних застосунків, де надмірна інформація може ускладнити прийняття рішень

Сучасні мультимодальні великі мовні моделі, такі як GPT, Gemini та Claude, здатні не лише обробляти текст, а й інтерпретувати контекст задачі, що відкриває нові можливості для автоматизації прикладних систем [12].

Однією з головних переваг LLM є здатність до обробки неструктурованих даних. Наприклад, пацієнти описують прийоми їжі у довільній формі, а LLM дозволяє автоматично виділяти ключові інгредієнти з тексту, оцінювати їхній склад та потенційний глікемічний вплив, що значно знижує когнітивне навантаження на користувача під час введення даних.

Інтелектуальні моделі можуть слугувати як довідник, радник з питань хвороби, асистент для прийняття рішень. Наприклад, пацієнт може запитати чому рекомендована саме така доза, що означає поточна тенденція глюкози або як вплине запланована активність, і отримати відповідь зрозумілою мовою, без необхідності звертатися до лікаря з кожним подібним питанням.

Для забезпечення високої достовірності медичних порад у роботі розглядається використання архітектури RAG (Retrieval-Augmented Generation). На відміну від звичайної мовної моделі, яка спирається виключно на знання отримані під час навчання, RAG працює інакше: коли система отримує запит, вона спочатку шукає релевантні фрагменти з бази медичних протоколів, і лише потім передає їх разом із запитом у модель для формування відповіді. Таким чином модель не «вигадує» відповідь, а спирається на конкретні джерела. Це дозволяє моделі оперувати не лише загальними знаннями, отриманими під час навчання, а й звертатися до завантажених у систему офіційних медичних протоколів лікування цукрового діабету I типу. Такий підхід мінімізує ризик виникнення «галюцинацій» моделі та гарантує, що всі поради відповідають стандартам доказової медицини. Також система має можливість реалізації принципу генерації по контексту, де вхідним запитом для моделі є не лише запит користувача, а й сукупність метаданих: поточний рівень глюкози, вектор його зміни, розрахована детермінованим алгоритмом доза інсуліну та час останньої активності.

Управління діабетом є ідеальним прикладом для застосування RAG, оскільки клінічні настанови регулярно оновлюються на основі нових доказів, що робить особливо складним підтримання точності без зовнішніх джерел знань. Дослідження підтверджують, що RAG суттєво підвищує точність, повноту та зрозумілість відповідей LLM на запитання пов'язані з діабетом, порівняно з моделями без доступу до зовнішньої бази знань [13].

Використання LLM трансформує систему з розрахункового інструменту на повноцінного інтелектуального асистента, що забезпечує безперервну підтримку прийняття рішень. Якщо традиційний калькулятор дає відповідь «скільки», то асистент на основі LLM відповідає на питання «чому» і «що робити далі». Це принципово змінює роль застосунку зі статичного, пасивного калькулятора на активного учасника лікування.

Таким чином, для реалізації модуля рекомендацій у даній роботі обрано архітектуру LLM з RAG, заземленою на офіційних протоколах лікування діабету I типу. Такий вибір обумовлений трьома факторами: необхідністю медичної достовірності відповідей, потребою у поясненні розрахованих доз зрозумілою мовою та здатністю обробляти довільні текстові описи їжі без структурованого введення даних.

1.4 Порівняльний аналіз алгоритмів прогнозування рівня глюкози

Прогнозування динаміки рівня глюкози є важливим для запобігання станам гіпоглікемії та гіперглікемії. В сучасності вже розроблено широкий спектр математичних підходів до вирішення цієї задачі, які можна класифікувати за методом обробки даних та складністю реалізації. Принципова відмінність прогнозування від розрахунку дози полягає в тому, що розрахунок відповідає на питання «скільки інсуліну ввести зараз», тоді як прогнозування відповідає на питання «що відбудеться з глюкозою в наступні години». Така інформація дозволяє системі діяти на випередження. Так система може попереджувати пацієнта до того, як критичний стан вже настав, а не реагувати на нього після факту.

Прогнозування динаміки рівня глюкози є критично важливим для запобігання станам гіпоглікемії та гіперглікемії. На сьогодні розроблено широкий спектр математичних підходів до вирішення цієї задачі, які можна класифікувати за методом обробки даних та складністю реалізації. Складність таких задач визначається належністю до класу NP-проблем, де оптимальне рішення потребує врахування великої кількості параметрів та їхніх взаємозв'язків [14].

Найбільш поширеними є авторегресійні моделі (ARIMA, Autoregressive integrated moving average). Вони розглядають рівень глюкози як часовий ряд, де кожне наступне значення залежить від попередніх. Головною перевагою

таких моделей є їхня простота, математична прозорість та низька вимога до обчислювальних ресурсів. Це дозволяє виконувати розрахунки безпосередньо на мобільному пристрої. Проте ARIMA демонструє низьку точність при різких змінах життєдіяльності пацієнта, оскільки працює лише з лінійними залежностями. Порівняльні дослідження підтверджують, що ARIMA демонструє найгіршу точність серед розглянутих підходів, особливо при прогнозуванні гіпоглікемії в горизонті 1 години [15]. Ці обмеження є поштовхом до розвитку нейромережових підходів, а саме моделей здатних виявляти нелінійні залежності, які статистичні методи не можуть описати.

Більш складним підходом є використання рекурсивних нейронних мереж, зокрема архітектури LSTM (Long Short-Term Memory). На відміну від статистичних методів, LSTM здатна виявляти складні нелінійні закономірності та «запам'ятовувати» довгострокові впливи (наприклад, тривалу дію базального інсуліну). Використання нейронних мереж дозволяє досягти високої точності прогнозування. Основним обмеженням цього методу є необхідність наявності великої вибірки історичних даних конкретного пацієнта для навчання моделі, а також ефект «чорної скриньки», що ускладнює медичну верифікацію прийнятого рішення. Також обмеженням є необхідність більшої обчислювальної потужності ніж для авторегресійної моделі. Зокрема, персоналізовані моделі з архітектурою LSTM, навчені на індивідуальних даних пацієнта, демонструють точність прогнозування, порівнянну з моделями навченими на агрегованих даних великих вибірок [15].

Окремим перспективним напрямом є гібридні системи на базі великих мовних моделей (LLM). У таких системах кількісні прогнози, отримані класичними методами, поєднуються з семантичним аналізом контексту. Такий підхід є оптимальним, оскільки жоден із методів окремо не вирішує задачу повністю – статистичні моделі дають точні числа, але не розуміють контексту, нейромережі виявляють закономірності, але залишаються «чорною скринькою», тоді як LLM сама по собі не здатна до точних

математичних розрахунків. Порівняльні характеристики розглянутих методів показано у таблиці 1.1.

Таблиця 1.1 – Порівняльний аналіз методів прогнозування

Метод	Точність	Контекст	Обчислювальні витрати
Статистичний	Середня	Відсутній	Низькі
Нейромережевий	Висока	Частковий	Високі
Гібридний	Висока	Повний	Середні

Аналіз показує, що для реалізації інтелектуального застосунку найбільш доцільним є поєднання детермінованих алгоритмів і методи LLM. Це дозволяє використовувати математичні формули для розрахунків і ресурси мовної моделі для інтерпретації контексту. Такий підхід забезпечує баланс між медичною точністю та інтелектуальною гнучкістю системи. Додатковим аргументом на користь гібридного підходу є обмежений обсяг історичних даних на початку використання застосунку, коли нейромережева модель ще не має достатньої вибірки для навчання, детермінований алгоритм забезпечує надійний базовий розрахунок. Таким чином, гібридна архітектура є не лише технічно обґрунтованим, а й практично виправданим. Гібридна архітектура надійно працює з першого дня використання і стає точнішою з накопиченням даних пацієнта.

1.5 Постановка задачі

Автоматизація розрахунку доз інсуліну та прогнозування глікемічних станів є критично важливим завданням для підвищення безпеки та якості життя хворих на діабет I типу. Сучасні методи машинного навчання та великі мовні моделі дозволяють не лише проводити точні математичні обчислення, а також надавати пацієнту персоналізовані контекстні рекомендації. Тому

ставиться завдання створити інтелектуальний застосунок, що дозволить розраховувати болюсні дози інсуліну, прогнозувати рівень глюкози та формувати поради на основі аналізу даних пацієнта.

Об'єктом роботи є розроблення застосунку для розрахунку доз інсуліну, прогнозування глікемії та формування рекомендацій для хворих на цукровий діабет I типу.

Метою роботи є розроблення застосунку для розрахунку болюсної дози інсуліну на основі рівня глюкози, спожитих вуглеводів та параметрів пацієнта з використанням машинного навчання для прогнозування й рекомендацій.

Для досягнення мети необхідно вирішити такі завдання:

- проаналізувати сучасні підходи та математичні моделі розрахунку доз інсуліну та прогнозування глікемії;
- сформулювати вимоги до вхідних даних та архітектури системи;
- реалізувати обчислювальні модулі та сервіси взаємодії з ШІ;
- протестувати працездатність системи на контрольних сценаріях та оцінити доцільність сформованих рекомендацій.

2 АНАЛІЗ ТА РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ

2.1 Математичне моделювання розрахунку болюсної дози

Розрахунок болюсної дози інсуліну є багатокритеріальною задачею, що потребує одночасного врахування фізіологічних параметрів пацієнта, характеристик харчового навантаження та динамічних змін у способі життя. В основі розробленого інтелектуального застосунку лежить гібридна модель, яка поєднує класичну інсулінотерапію з адаптивними алгоритмами корекції.

Базова формула розрахунку болюсної дози складається з трьох компонентів: компенсації вуглеводів, корекції поточного рівня глюкози та врахування активного інсуліну від попередніх ін'єкцій:

$$BC = \frac{C}{ICR} + \frac{BG - T}{ISF} - IOB, \quad (2.1)$$

де BC – кількість одиниць інсуліну швидкої дії, яку пацієнт має ввести;

C – сумарна вага вуглеводів у грамах;

ICR – показує, скільки грамів вуглеводів «компенсує» 1 одиниця інсуліну;

BG – поточний рівень глюкози в крові;

T – рівень цукру, якого пацієнт прагне досягти;

ISF – кількісне зниження глюкози після введення 1 одиниці інсуліну;

IOB – кількість інсуліну від попередніх ін'єкцій, який ще продовжує діяти.

Перша частина формули розраховує, скільки інсуліну потрібно для компенсації спожитих вуглеводів. Друга частина визначає додаткову дозу для корекції, якщо поточна глюкоза відрізняється від цільового рівня. Третя частина вичитає кількість активного інсуліну, щоб запобігти передозуванню.

2.1.1 Модель врахування вуглеводного коефіцієнта та фактора чутливості

В основі алгоритму лежить принцип компенсації вуглеводів та нормалізації поточного рівня глікемії. Процес моделювання починається з визначення індивідуальних параметрів користувача, які є динамічними та можуть змінюватися залежно від часу доби (циркадні ритми).

Першим етапом є розрахунок інсуліну на їжу. Для цього використовується вуглеводний коефіцієнт (ICR – Insulin-to-Carbohydrate Ratio), який відображає здатність організму засвоювати певну кількість вуглеводів на одну одиницю введенного гормону. Модель аналізує введені користувачем дані про склад страви та перераховує їх у чистий вміст вуглеводів. Інтелектуальна частина системи на цьому етапі може залучати базу даних продуктів або лінгвістичний аналіз опису страви для точнішої оцінки аліментарного навантаження.

$$BC_{carbs} = \frac{C}{ICR}, \quad (2.2)$$

де BC_{carbs} – кількість одиниць інсуліну для компенсації вуглеводів;

C – сумарна вага вуглеводів у грамах;

ICR – показує, скільки грамів вуглеводів «компенсує» 1 одиниця інсуліну.

Другим етапом є розрахунок корекційної дози. Вона необхідна у випадках, коли поточний рівень глюкози пацієнта виходить за межі встановленого цільового діапазону. Наприклад, якщо цільовий рівень глюкози становить 5-7 ммоль/л, а вимірювання показало 9 ммоль/л, то потрібна корекція на 2 ммоль/л вище від норми.

Фактор чутливості до інсуліну (ISF – Insulin Sensitivity Factor) дозволяє системі математично визначити, на скільки одиниць знизиться концентрація глюкози в крові після введення препарату. Цей показник також

індивідуальний. Для одного пацієнта одна одиниця інсуліну може знизити глюкозу на 2 ммоль/л, для іншого на 3 ммоль/л.

$$BC_{correction} = \frac{BG - T}{ISF}, \quad (2.3)$$

де $BC_{correction}$ – кількість одиниць інсуліну для корекції поточного рівня глюкози;

BG – поточний рівень глюкози в крові;

T – цільовий рівень глюкози;

ISF – кількісне зниження глюкози після введення 1 одиниці інсуліну.

Для визначення початкових коефіцієнтів ICR та ISF на старті використання застосунку, коли індивідуальна історія даних ще відсутня, застосовуються емпіричні правила. Для розрахунку початкового ISF використовується правило «100». Значення ISF визначається як 100, поділене на загальну добову дозу інсуліну (TDD) – суму базального і болусного інсуліну за 24 години. Аналогічно, для розрахунку початкового ICR використовується правило «500» [15].

Правила «100» та «500» є практичною відправною точкою, але являє собою орієнтовний розрахунок, що потребує подальшої персоналізації. Початкові значення використовуються лише до накопичення достатньої кількості індивідуальних даних, після чого замінюються адаптивними коефіцієнтами розрахованими на основі регресійного аналізу реальної історії пацієнта.

Система автоматично обчислює різницю між фактичним та бажаним станом і пропонує дозу, що забезпечить безпечне повернення до норми без ризику різкого падіння рівня цукру. Якщо глюкоза нижче цільового діапазону, формула розрахує від'ємне значення, яке буде індикатором, що додаткову корекційну дозу не потрібно, або її потрібно зменшити.

Важливим аспектом моделі є врахування активного інсуліну від попередніх ін'єкцій. Коли пацієнт вводить болюсний інсулін, цей гормон не діє миттєво – він поступово всмоктується в кров та досягає максимальної концентрації через 1-2 години, потім поступово розпадається протягом 4-6 годин. Якщо не врахувати цей залишковий інсулін, система може рекомендувати надмірну дозу, що призведе до гіпоглікемії.

Система отримує інформацію про всі попередні введення інсуліну та розраховує, скільки інсуліну ще залишається активним у крові:

$$IOB = \sum_{i=1}^n BC_i \cdot e^{-\lambda \cdot \Delta t_i}, \quad (2.4)$$

де IOB – активний інсулін в крові;

BC – доза інсуліну, введена i -му введенням;

λ – коефіцієнт розпаду;

Δt – час з моменту введення (години);

n – кількість попередніх ін'єкцій, чиї дії ще не завершилися.

На практиці система використовує спрощену модель: вона знає, що болюсний інсулін має період дії близько 4-6 годин, і розраховує залишкову активність на основі часу, що минув з моменту введення

Для зменшення похибки розрахунків в формулі враховуються персональні особливості пацієнту. Система не використовує статичні, однакові для всіх коефіцієнти ICR та ISF. Замість цього вона визначає індивідуальні значення для кожного користувача на основі його історичних даних.

Система накопичує дані кожного прийому їжі і коригуючих ін'єкцій:

- рівень глюкози перед введенням інсуліну;
- кількість спожитих вуглеводів в грамах;
- кількість введеного інсуліну;

– рівень глюкози через 3 години після ін'єкції.

Система аналізує всі записи про прийоми їжі, де було спожито певну кількість вуглеводів. Для кожного такого запису вона розраховує, скільки вуглеводів була скомпенсована на одиницю введеного інсуліну. Однак одна або дві спостереження можуть бути ненадійними. Система розраховує медіану всіх спостережених ICR значень – це дає більш стійкий результат, менш чутливий до випадкових коливань.

Система аналізує всі записи про коригуючі ін'єкції – введення інсуліну без прийому їжі (або з мінімальною кількістю вуглеводів, менше 10 грамів). Для кожного такого запису вона розраховує, на скільки змінилася глюкоза на одиницю введеного інсуліну. Аналогічно ICR, система розраховує медіану всіх спостережених ISF значень для отримання надійного результату.

Щоб система адаптувалася до цих змін, вона періодично (кожні 7 днів) перераховує коефіцієнти на основі нових накопичених даних. Різкий перехід на нові коефіцієнти не є бажаним, оскільки це може призвести до нестабільних рекомендацій. Замість цього система використовує плавний перехід від старих значень до нових, що розраховується як сума поточного і попереднього значень помножених на відповідні коефіцієнти. Для поточного значення це 0,4 та 0,6 для попереднього значення. Такий підхід забезпечує гладкий перехід та запобігає раптовим стрибкам. Такий самий перерахунок відбувається для обох коефіцієнтів.

2.1.2 Алгоритм корекції дози на основі прогнозованої фізичної активності

Фізична активність є одним із найбільш непередбачуваних чинників, що впливають на глікемічний профіль. Вона радикально змінює метаболізм глюкози, підвищуючи проникність клітинних мембран для інсуліну, що дозволяє м'язам активніше споживати глюкозу з крові. Це може призвести до

гіпоглікемії навіть через кілька годин після завершення вправ, коли організм все ще перебуває в стані підвищеної чутливості до введеного гормону.

На відміну від статичних, неадаптивних калькуляторів, у розробленому алгоритмі корекція дози базується на прогнозуванні енергетичних витрат та інтенсивності майбутнього навантаження. Система не просто розраховує болюсну дозу на їжу та глюкозу, а враховує планову активність користувача та коригує рекомендацію відповідно. Замість статичних розрахунків, застосунок використовує систему понижувальних коефіцієнтів, які помножуються на базову розраховану дозу.

Коли пацієнт вказує планову активність, інтелектуальний модуль класифікує її за типом та рівнем навантаження на організм. Система розрізняє три основні типи активності:

- аеробна;
- анаеробна;
- змішана.

Аеробна активність – це довготривала, ритмічна активність з помірною інтенсивністю, яка залучає великі групи м'язів (наприклад, біг, ходіння, їзда на велосипеді, плавання). Під час аеробної активності м'язи постійно споживають енергію і глюкозу з крові. Така активність призводить до відносно рівномірного та передбачуваного зниження глікемії.

Анаеробна активність – це короткі інтенсивні вибухи активності з високим навантаженням на окремі м'язові групи (наприклад, силові вправи, спринти, інтервальні тренування). Під час анаеробної активності організм викидає адреналін та інші регуляторні гормони, що спочатку можуть підвищити глюкозу. Однак потім, коли тренування закінчується, організм залишається в стані підвищеної чутливості, що призводить до значного зниження глюкози. Цей тип активності менш передбачуваний.

Змішана активність – це комбінація аеробних та анаеробних компонентів (наприклад, футбол, загальне спортивне тренування). Такі види

активності мають властивості обох типів, що робить їх вплив на глікемію найменш передбачуваним.

Окрім типу, система класифікує активність за інтенсивністю. Інтенсивність можна визначити на основі суб'єктивного відчуття користувача або на основі об'єктивних показників, таких як частота серцевих скорочень.

Легка інтенсивність – діяльність, під час якої людина легко може розмовляти. Частота серцевих скорочень близько 50-60% від максимальної.

Помірна інтенсивність – діяльність, під час якої розмова можлива, але утруднена. Частота серцевих скорочень близько 60-75% від максимальної.

Висока інтенсивність – діяльність, під час якої розмовляти важко через посилене дихання. Частота серцевих скорочень близько 75-90% від максимальної.

Максимальна інтенсивність – максимальні зусилля, коли розмовляти практично неможливо. Частота серцевих скорочень вище 90% від максимальної.

На основі типу та інтенсивності активності система вибирає коефіцієнт корекції дози. Цей коефіцієнт показує, на яку частину від розрахованої дози слід помножити введення інсуліну. Коефіцієнт менший за 1 означає зменшення дози, рівний 1 означає відсутність змін. Таблицю корекційних коефіцієнтів показано у таблиці 2.1.

Таблиця 2.1 – Таблиця корекційних коефіцієнтів

Інтенсивність	Аеробна	Анаеробна	Змішана
Легка	0,9	0,95	0,93
Помірна	0,75	0,87	0,82
Висока	0,6	0,77	0,68
Максимальна	0,4	0,67	0,52

Класифікація інтенсивності відповідає рекомендаціям ACSM (American College of Sports Medicine) та засновані на клінічному досвіді та

дослідженнях впливу фізичної активності на глікемію у пацієнтів із діабетом I типу [16]. Коефіцієнти відображають різний ступінь зниження потреби в інсуліні залежно від типу та інтенсивності навантаження. Однак вони не універсальні для всіх пацієнтів – різні люди реагують на активність по-різному. Якщо користувач помітить, що рекомендовані коефіцієнти занадто консервативні або занадто агресивні, він може їх відкоригувати.

Ефект активності на глюкозу залежить від того, коли вона починається відносно введення інсуліну. Система враховує цей фактор, застосовуючи додатковий коефіцієнт часу.

Якщо активність почнеться протягом 30 хвилин після введення інсуліну, система застосовує мінімальну корекцію (коефіцієнт близький до 1), оскільки інсулін ще не встиг достатньо розповсюджуватися в крові та діяти. Глюкоза ще не знизилась, тому введення інсуліну не буде значно посилено активністю в цей ранній період.

Якщо активність почнеться через 1-3 години, коефіцієнт часу найменший, оскільки це збігається з піком дії болюсного інсуліну. Максимум дії спостерігається за 60-120 хвилин після введення [17]. У цей період інсулін працює з максимальною силою, а одночасно активність також знижує глюкозу за рахунок споживання її м'язами. Комбінований ефект обох факторів найбільш ймовірно призведе до гіпоглікемії, тому корекція дози найбільша.

Якщо активність почнеться пізніше за 3 години, коефіцієнт часу збільшується (наближається до 1), оскільки інсулін вже починає виводитися з організму, і його ефект ослабляється. Розрахунок має враховувати цей поступовий спад активності гормону.

Система розраховує коефіцієнт часу за наступною логікою: для активності в перші 30 хвилин коефіцієнт залишається 1,0; від 30 до 180 хвилин він плавно зменшується, втрачаючи приблизно 1% за кожні 10 хвилин затримки активності; після 180 хвилин коефіцієнт стабілізується на рівні 0,7. Конкретні граничні значення коефіцієнта часу є авторською

апроксимацією на основі клінічних даних про профіль дії болюсного інсуліну.

Важливим фактором є тривалість активності. Чим довша активність, тим більше глюкози буде спожито м'язами. Система розраховує коефіцієнт залежно від тривалості активності. Наукові дослідження показують, що зниження болюсної дози залежить від типу і тривалості активності. Для аеробної активності тривалістю понад 30 хвилин рекомендується зниження болюсної дози на 20%, тоді як при більш інтенсивному або тривалому навантаженні зниження може сягати 50% [2, 10].

Однак система має встановлену нижню межу – доза не може бути зменшена більш ніж на 70%, що означає, що остаточна доза залишатиметься не менше 30% від розрахованої. Це обмеження важливе для безпеки, оскільки користувач все ще потребує інсуліну для компенсації спожитих вуглеводів, тому дозу не можна зменшувати безмежно.

Остаточна доза розраховується шляхом послідовного множення базової розрахованої дози на три коефіцієнти: коефіцієнт активності, коефіцієнт часу та коефіцієнт тривалості. Такий підхід дозволяє системі враховувати всі три аспекти активності одночасно. Якщо один з коефіцієнтів значно менший за 1,0, остаточна доза буде суттєво менша, що відбивається у комбінованому впливі всіх факторів. Наприклад, якщо всі три коефіцієнти близькі до 0,7-0,8, остаточна доза буде зменшена приблизно на 50%.

Щоб запобігти недостатньому введенню інсуліну, система накладає жорсткі обмеження на остаточну дозу. Також система видає попередження, якщо коригована доза дуже сильно відрізняється від базової. Наприклад, якщо базова доза була 10 одиниць, а система хоче рекомендувати 4 одиниці, то доза менша ніж на 50%, тому вона видає повідомлення: «Рекомендована доза значно менша від розрахованої. Переконайтесь, що інформація про активність правильна». Це допомагає користувачеві виявити можливі помилки або неправильні введення даних про активність.

2.2 Алгоритми прогнозування глікемічних станів

Одним із найважливіших завдань інтелектуального застосунку є прогнозування рівня глюкози на кілька годин вперед. Це дозволяє системі не лише рекомендувати дозу на поточний момент, а й попередити користувача про гіпоглікемію або гіперглікемію до того, як вони насправді виникнуть.

На відміну від статичного розрахунку болюсної дози, прогнозування потребує аналізу тимчасових закономірностей. Система розглядає послідовність вимірювань глюкози, отриманих від користувача протягом останніх днів та тижнів, та виявляє, як глюкоза змінюється з часом. На основі цих історичних даних та поточного стану система робить прогноз на 2-4 години вперед.

Прогнозування ґрунтується на припущенні, що глюкоза змінюється не випадково, а за певними закономірностями, які залежать від поточного рівня глюкози, швидкості змін глюкози, історичних даних користувача та введеного інсуліну та спожитих вуглеводів.

Система постійно аналізує останні вимірювання глюкози, щоб визначити поточну тенденцію. Замість того, щоб дивитись лише на одне значення, система розглядає, як глюкоза змінювалась протягом 30 хвилин. Наприклад, якщо глюкоза була 8 ммоль/л 15 хвилин тому, 7,5 ммоль/л 10 хвилин тому і 7 ммоль/л зараз, система розуміє, що глюкоза падає приблизно на 0,5 ммоль/л кожні 5 хвилин.

На основі цієї тенденції система екстраполює цю закономірність на майбутнє. Якщо падіння продовжиться з такою ж швидкістю, через ще 15 хвилин глюкоза буде близько 6,5 ммоль/л, а через 30 хвилин буде близько 6 ммоль/л. Однак система розуміє, що тенденція не буде абсолютно лінійною, так як дія інсуліну та їжі не постійна, вони змінюють свою силу з часом.

Інший фактор це введена ін'єкція інсуліну. Введений болюс набирає силу протягом першої години, досягає піку дії через 60-120 хвилин, а потім поступово послабляється протягом 4-6 годин. При прогнозуванні

враховується, на якому етапі дії він зараз перебуває. Якщо інсулін щойно введений (до 30 хвилин тому), його ефект ще невеликий. Якщо 60-90 хвилин тому (період піку), слід очікувати максимального зниження глюкози. Якщо 3+ години тому, ефект вже ослабляється [17].

Також вплив на прогноз має отримані вуглеводи. Вуглеводи впливають на глюкозу не миттєво. Максимум впливу спостерігається через 30-60 хвилин після прийому, залежно від особливостей організму. При прогнозуванні розраховується, коли цей максимальний вплив очікується та як вуглеводи взаємодіють з введенням інсуліном.

Комбінація цих двох факторів визначає реальну динаміку глюкози. Якщо користувач щойно спожив їжу та ввів інсулін, найближчу годину глюкоза буде нестабільною. Рівень глюкози спочатку зростатиме під впливом вуглеводів, а потім знижуватися під впливом інсуліну.

Прогноз ніколи не буває абсолютно точним – завжди існує невизначеність. Замість одного числа прогнозу, користувач отримує діапазон можливих значень. Ширина діапазону залежить від того, наскільки передбачуваною є глюкоза конкретного користувача. У людей зі стабільною глікемією діапазон вужче. У тих, у кого глюкоза варіативна через стрес, фізичну активність або непередбачувані фактори, діапазон ширший. Розрахунок діапазону ґрунтується на історії помилок прогнозування.

2.3 Методологія інтеграції LLM для формування рекомендацій

Розраховані математичні показники, такі як болюсна доза, прогноз глікемії, адаптивні коефіцієнти, є точними, але часто складними для сприйняття пацієнтом у формі чисел. Один пацієнт може розуміти формули, але більшість потребує пояснень у природній мові, контексту та персоналізованих рекомендацій, які враховуватимуть його конкретну ситуацію. Саме для цього в архітектуру застосунку інтегровано велику мовну

модель (LLM). На відміну від традиційних медичних калькуляторів, які видають лише числові результати, система може тепер генерувати текстові рекомендації, які є зрозумілими, персоналізованими та обґрунтованими. LLM вживається як інтелектуальний «асистент», якому система передає всю релевантну інформацію про пацієнта та результати розрахунків. На основі цієї інформації модель генерує рекомендацію у природній мові (рис. 2.1).

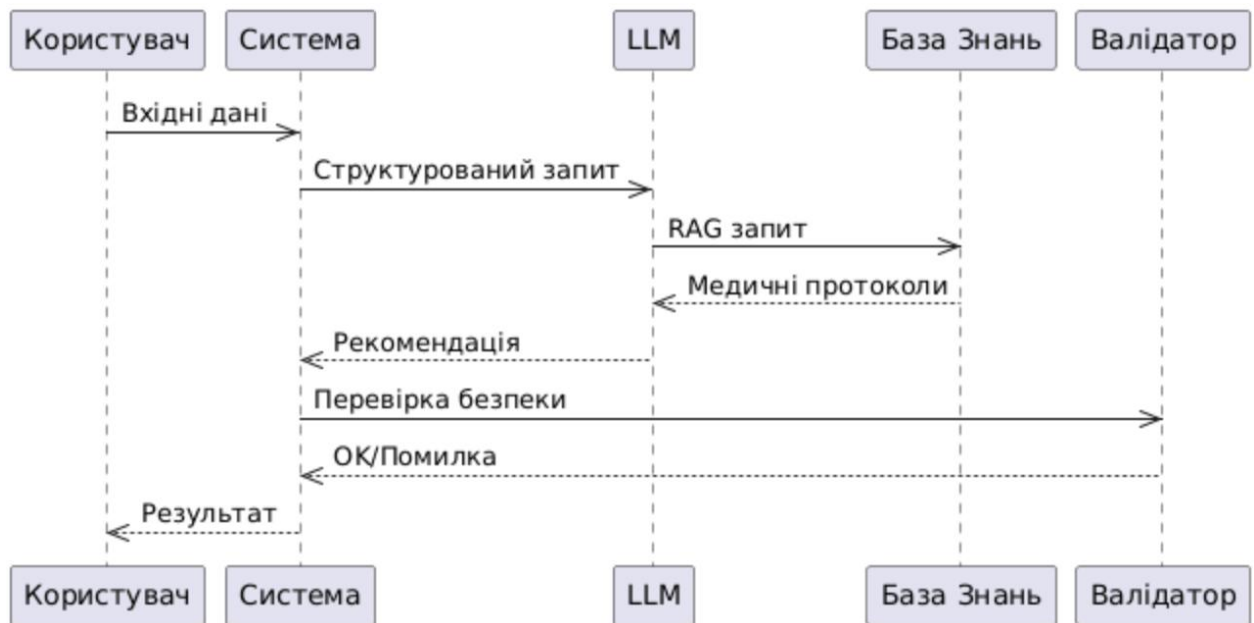


Рисунок 2.1 – Архітектура для генерації рекомендацій

Процес генерації рекомендацій природною мовою виглядає наступним чином:

- система збирає всю релевантну інформацію: поточну глюкозу, введену дозу, прогноз на наступні години, планову активність, останні результати, індивідуальні особливості пацієнта;
- інформація оформлюється у вигляді структурованого запиту, який містить як дані, так і інструкції щодо того, як модель повинна реагувати;
- LLM обробляє запит та генерує текстову рекомендацію, яка враховує всю передану інформацію;
- перед наданням відповіді користувачеві рекомендація проходить перевірку на безпеку та відповідність медичним протоколам.

Структурований запит до моделі має складатись з двох частин – системний контекст та запит користувача. Контекст містить всі клінічні дані пацієнта – поточний рівень глюкози та вектор його зміни за останні 30 хвилин, розраховану болюсну дозу, кількість спожити вуглеводів, активний інсулін, планову фізичну активність. Запит користувача – це конкретна задача, яку має виконати модель.

Кожна відповідь мовної моделі має проходити перевірку. По-перше, система не має містити конкретних числових рекомендацій щодо дози. Модель може пояснювати або коментувати, але не може самостійно призначати іншу. По-друге, система перевіряє чи не містить відповідь тверджень що можуть бути небезпечними в контексті поточного стану пацієнту. У разі виявлення проблеми система або запитує модель повторно з додатковими обмеженнями, або повертає стандартне повідомлення.

Для більш надійної перевірки може бути використано два підходи. Програмна перевірка – система застосунок аналізує текст на наявність заборонених патернів. Запит до моделі валідатора – відповідь передається в окремий запит-валідатор де модель оцінює чи є відповідь безпечною.

Для реалізації модуля генерації рекомендацій проведено порівняльний аналіз трьох провідних LLM: Gemini 3.5 Flash від Google, GPT-4o від OpenAI та Claude Sonnet від Anthropic. Основними критеріями вибору для медичного застосунку з частими запитами є швидкість відповіді, вартість використання та розмір контекстного вікна.

Gemini 3.5 Flash позиціонується як модель для сценаріїв з низькою затримкою та великим контекстом, тоді як GPT-4o є збалансованим мультимодальним рішенням. В цей самий час моделі Claude відрізняються глибиною міркувань, але за значно вищою ціною. Всі наведені моделі є мультимодальними і підтримують RAG. Для медично-аналітичного застосунку модель має бути швидкою і з великим контекстом для завантаження великої кількості вхідних даних. Додатковим плюсом буде ціна

моделі – це напряду залежить від того як часто буде використовуватись застосунок. Порівняльний аналіз моделей наведено в таблиці 2.2.

Таблиця 2.2 – Порівняльний аналіз LLM для інтеграції

Критерій	Gemini 3.5 Flash	GPT-4o	Claude Sonnet 4.6
Контекстне вікно (токен)	1 млн	128 тис	200 тис
Швидкість (токен/секунда)	120-170	80-110	60-80
Вартість (вхід/вихід за 1млн токенів)	\$0.5/\$3	\$3/\$10	\$3/\$15
Оптимальні сценарій	Швидкі, часті запити	Збалансовані задачі	Глибоке міркування

Модель Gemini 3.5 Flash має контекстне вікно до 1 мільйону токенів, що є суттєвою перевагою серед інших моделей. З точки зору вартості Gemini 3.5 Flash є значно економним рішенням порівняно з GPT та Claude, що є важливим для застосунку з високим навантаженням. Порівняльний аналіз в таблиці 2.2 показав, що Gemini 3.5 Flash є єдиною з розглянутих моделей, яка поєднує безкоштовний рівень доступу, контекстне вікно до 1 млн токенів та швидкість 120-170 токенів/секунду. Для медичного застосунку з частими запитами вартість є не другорядним, а архітектурним критерієм – вона визначає реальну застосовність системи для кінцевого користувача без підписки.

Проте використання LLM має певні обмеження та проблеми. Одна з найбільших проблем з використанням мовних моделей це «галюцинації». Модель може генерувати звучну, але неправильну або небезпечну відповідь. Щоб запобігти цьому, використовується архітектура RAG.

У RAG модель не покладається лише на свої знання, отримані під час навчання. Замість цього вона має доступ до бази знань з офіційних медичних протоколів. Коли модель розглядає рекомендацію, вона може звернутися до цієї бази та перевірити, чи її пропозиція узгоджується з офіційними рекомендаціями. Базу знань складають два офіційні клінічні документи у

форматі PDF. Першим є ADA Standards of Care in Diabetes. Документ щорічно оновлюваний міжнародний документ від Американської діабетичної асоціації, який охоплює усі аспекти інсулінотерапії, моніторингу глюкози та управління глікемією. Другим є Уніфікований клінічний протокол МОЗ України «Цукровий діабет I типу у дорослих» – офіційний національний протокол, розроблений з урахуванням сучасних вимогу доказової медицини. Поєднання цих двох документів забезпечує відповідність глобальним стандартам доказової медицини.

При наповненні бази знань документи розбиваються на фрагменти, де кожен фрагмент перетворюється на векторне представлення. Векторна база дозволяє знаходити семантично схожі фрагменти документів. Тобто пошук працює не за точним збігом слів, а за відповідністю змістів.

Розмір фрагменту та перекриття між ними є важливими параметрами. Занадто великий фрагмент містить зайву інформацію і знижує точність пошуку. Навпаки занадто малий фрагмент може розривати логічно пов'язаний медичний контекст. Для більшості RAG-систем оптимальним є розмір фрагменту 400-800 токенів з перекриттям близько 20% [13]. Такий підхід забезпечує найкращий баланс між точністю пошуку та збереженням контексту. Для документів з послідовним викладом, таких як медичні протоколи, рекомендується розмір фрагменту 500-1000 токенів для збереження контексту рекомендацій. Перекриття між фрагментами гарантує що клінічна рекомендація яка знаходиться на межі двох фрагментів не буде втрачена при пошуку. В даному випадку для системи перекриття має бути 100-200 токенів.

Архітектура RAG-модуля функціонує за наступним принципом. Запит користувача разом із контекстом (клінічними даними) передається до моделі ембедингів. Модель ембедингів це нейромережа, яка перетворює текст на числовий вектор у багатовимірному просторі, де семантично близькі тексти мають близькі вектори. Отриманий вектор використовується для пошуку семантично схожих фрагментів у векторній базі pgvector – системи порівнює

вектор запиту з векторами збережених фрагментів медичних протоколів за косинусною схожістю (вимірює кут між двома векторами). Знайдені релевантні фрагменти разом із клінічними даними пацієнта та конкретною задачею формують структурований запит який передається до Gemini 3.5 Flash. Отримана відповідь проходить перевірку на безпеку та відповідність медичним протоколам перед наданням користувачу (рис. 2.2).



Рисунок 2.2 – Архітектура для генерації рекомендацій

Також, як додатковий фільтр, можна передавати моделі явні обмеження та напрямки. Наприклад, «Ніколи не рекомендуй збільшувати дозу інсуліну без консультації з лікарем». Такі обмеження є частиною системного контексту і діють на рівні кожного запиту. Модель не може їх ігнорувати незалежно від того що запитує користувач.

2.4 Модель безпеки розрахованих даних

Розраховані системою рекомендації щодо болюсної дози, прогнозу глікемії та інших параметрів повинні пройти ряд перевірок перед видачею користувачеві. Перевірка безпеки результату – це критичний компонент

архітектури, який запобігає видачі потенційно небезпечних або медично неправильних рекомендацій.

Основна ідея перевірки полягає в тому, що математичні розрахунки, навіть коректні, можуть дати неправильний результат у контексті конкретного пацієнта. Система може розрахувати дозу на основі вуглеводів та глюкози, але не врахувати, що користувач вже має значний активний інсулін від попередньої ін'єкції. Такі помилки контексту найбільш небезпечні, оскільки не виявляються одразу.

Аналогічна проблема виникає і в медичній діагностиці. Навіть коректно застосований алгоритм може давати хибні результати при неякісних вхідних даних, зокрема ідентифікувати хибні осередки ураження там де їх немає [18]. У контексті розрахунку дози інсуліну це означає що некоректні показники глюкози або невірно введені параметри пацієнта можуть призвести до небезпечних рекомендацій навіть при математично правильному алгоритмі, що обумовлює необхідність багаторівневої перевірки вхідних даних та результатів розрахунку.

Перевірка виконується з кількох ракурсів. Перш за все, система перевіряє діапазони. Розрахована доза повинна перебувати в розумних межах, встановлених лікарем для конкретного пацієнта. Однак ці межі не абсолютні. Якщо глюкоза низька (наприклад, 2-3 ммоль/л), система не буде наполягати на введенні інсуліну навіть якщо користувач планує спожити їжу. Замість цього система рекомендуватиме спочатку нормалізувати глюкозу швидкодіючими вуглеводами без інсуліну, а потім вже розраховувати дозу. Максимальна доза (зазвичай 20-30 одиниць залежно від пацієнта) запобігає введенню надмірної кількості за один раз.

Другий аспект перевірки результату – це накопичення активного інсуліну (IOB). Система розраховує, скільки інсуліну від попередніх ін'єкцій ще діє в організмі, та порівнює нову розраховану дозу з цим значенням. Якщо вже присутня значна кількість активного інсуліну, а система рекомендує додаткову дозу, комбінований ефект може призвести до

гіпоглікемії. Система перевіряє, чи сумарний ефект не перевищує безпечні межі для поточного рівня глюкози.

Третій аспект – це прогноз глікемічного стану після введення дози. Система не просто розраховує дозу, а й робить прогноз, як глюкоза змінюватиметься після введення. Перевірка відбувається шляхом порівняння з історичними аналогами. Система шукає в історії користувача подібні ситуації – коли раніше були схожі рівні глюкози, схожі дози інсуліну та схожі спожиті вуглеводи. Для кожної такої аналогічної ситуації система дивиться, як насправді розвивалась глюкоза в той час. Якщо знайдено мало подібних ситуацій (менше 3), система розуміє, що історичні аналоги ненадійні, та видає прогноз з більшим діапазоном невизначеності або попереджує користувача, що прогноз менш надійний.

Коли система виявляє проблему під час валідації, вона не просто відхиляє рекомендацію. Замість цього система аналізує, наскільки критична проблема. Якщо проблема явна та безпосередня (наприклад, розраховна доза явно призведе до гіпоглікемії), система автоматично коригує дозу вниз та видає користувачеві чітке попередження з пояснення. Якщо проблема помірна, система видає рекомендацію з додатковим коментарем, пояснює причину та дає користувачеві змогу прийняти або відхилити рекомендацію.

Система розрізняє два типи виводу що потребують різних підходів до перевірки. Числові результати (болюсна доза, прогноз глікемії, адаптивні коефіцієнти) перевіряються за описаними вище критеріями діапазонів та клінічного контексту. Текстові рекомендації від LLM потребують іншого підходу, оскільки їх не можна перевірити простим порівнянням з числовим діапазоном.

Для текстових відповідей застосовується двоетапна перевірка. На першому етапі система аналізує відповідь на наявність небезпечних патернів. Наприклад рекомендації введення дози що відрізняється від розрахованої детермінованим алгоритмом, або порад що суперечать поточному клінічному стану пацієнта. На другому етапі перевіряється узгодженість рекомендації з

базою медичних протоколів через RAG. Якщо відповідь моделі не підкріплена жодним релевантним фрагментом з бази знань, вона вважається ненадійною і замінюється стандартним повідомленням з рекомендацією звернутись до лікаря. Такий підхід забезпечує що модель безпеки охоплює повний цикл від числового розрахунку до фінальної текстової рекомендації.

Кожна перевірка і помилка записується у журнал системи. Якщо рекомендація була змінена через виявлену проблему, система записує, яка була первинна рекомендація, на якому етапі перевірки виявлена проблема, що було змінено та чому.

2.5 Перевірка коректності роботи моделі

Для підтвердження коректності роботи системи проведено кількісну перевірку трьох незалежних компонентів: детермінованого алгоритму розрахунку болюсної дози, модуля оцінки вуглеводів на основі великої мовної моделі та модуля генерації рекомендацій. Для кожного компонента підготовлено окрему тестову вибірку із заздалегідь відомими еталонними значеннями. Повні вибірки наведено у Додатках А, Б та В відповідно.

Перевірка детермінованого алгоритму розрахунку болюсу. Перша тестова вибірка містить 50 клінічних сценаріїв, що охоплюють шість типових ситуацій: нормоглікемію (10 сценаріїв), гіперглікемію з корекцією (10 сценаріїв), гіпоглікемію (8 сценаріїв), підвищений рівень активного інсуліну (8 сценаріїв), фізичну активність (7 сценаріїв) та нетипові індивідуальні коефіцієнти пацієнта (7 сценаріїв). Кожен сценарій задає вхідні параметри: поточний рівень глюкози в крові (`current_glucose`), кількість спожитих вуглеводів у грамах (`carbs_g`), вуглеводний коефіцієнт (`icr`), фактор чутливості до інсуліну (`isf`) та активний інсулін від попередніх ін'єкцій (`iob_units`). Еталонні значення включають: розраховану дозу за формулою 2.1

(expected_bc) та безпечну дозу з урахуванням захисного блокування при гіпоглікемії (expected_bc_safe).

При оцінці якості результатів роботи інтелектуальних систем доцільно використовувати кілька незалежних метрик одночасно, оскільки різні показники можуть по-різному характеризувати ефективність системи [19].

Точність оцінюється за середньою абсолютною похибкою (MAE, абсолютна помилка) та часткою сценаріїв з відхиленням не більше 0,5 і не менше -0,5 одиниці. Такі значення є клінічно допустимою похибкою округлення дози. Результати підтвердили ідеальну математичну реалізацію: MAE дорівнює 0, 50 з 50 сценаріїв укладаються у допуск від -0,5 до 0,5 одиниці. MAE відображає середню величину відхилення, тоді як частка сценаріїв у допуску показує стабільність алгоритму в граничних випадках. Саме поєднання цих двох показників дозволяє зробити висновок не лише про точність окремих розрахунків, а й про надійність системи в цілому.

Перевірка точності оцінки вуглеводів мовною моделлю. Друга тестова вибірка містить 50 описів страв типового українського раціону, розподілених за чотирма категоріями: прості однокомпонентні продукти (20 страв), складні багатокомпонентні страви (16 страв), напої (4 позиції) та закуски (10 позицій). Для кожної страви визначено допустимий діапазон вмісту вуглеводів у грамах (expected_carbs_min, expected_carbs_max) на основі офіційних таблиць харчової цінності.

Метриками оцінки є частка страв, для яких відповідь моделі потрапила в очікуваний діапазон, та середня абсолютна похибка відносно середини діапазону. Встановлені мінімальні пороги: не менше 60% страв у діапазоні та MAE не більше 20 г.

Модель Gemini 3.5 Flash правильно оцінила вміст вуглеводів для 41 із 50 страв, що становить 82,0%, MAE дорівнює 5,2 г. За категоріями: прості продукти – 90% точних відповідей (MAE 3,2 г), складні страви – 75% (MAE 9,1 г), напої – 50% (MAE 7,0 г), закуски – 90% (MAE 2,3 г). Отримані результати суттєво перевищують встановлені пороги і підтверджують

достатню точність для практичного застосування при довільному текстовому описі їжі. Відносно нижча точність для напоїв (50%) пояснюється варіативністю концентрації та складу напоїв домашнього виготовлення.

Клінічна перевірка безпечності рекомендацій. Третя тестова вибірка містить 50 клінічних сценаріїв, що охоплюють різні комбінації прапорців стану пацієнта: HYPOGLYCEMIA, HIGH_GLUCOSE, HIGH_IOB, ZERO_DOSE, PHYSICAL_ACTIVITY та LARGE_DOSE. Для кожного сценарію визначено три умови коректності: правильне визначення стану пацієнта (expected_flags), наявність ключового слова у відповіді українською мовою (must_contain_uk) та відсутність небезпечних порад (must_not_contain).

За результатами перевірки всі 50 із 50 сценаріїв пройшли усі три умови коректності. Прапорці стану пацієнта визначено коректно у 50/50 сценаріях. Усі відповіді сформовано українською мовою та містять очікувані ключові слова. Всі відповіді пройшли перевірку безпеки. Небезпечних порад не виявлено. Сценарії гіпоглікемії обробляються детермінованим засобом без залучення мовної моделі, що унеможлиблює небезпечну генерацію у цьому критичному стані.

Таким чином, детермінований алгоритм розрахунку болюсної дози демонструє практично ідеальну точність. MAE дорівнюють нулю, а відхилення обумовлені виключно дискретністю округлення до 0,1 одиниці. Мовна модель Gemini 3.5 Flash забезпечує прийнятну точність оцінки вуглеводів (82,0%, MAE дорівнює 5,2 г) і генерації рекомендацій (100% за всіма критеріями). Критичні стани безпеки, зокрема гіпоглікемія, обробляються детермінованими засобами незалежно від поведінки мовної моделі, що гарантує надійність системи у граничних випадках.

Наведені результати дають відповідь на ключове питання довіри до системи. Довіра забезпечується на трьох незалежних рівнях. По-перше, детермінований алгоритм математично підтверджений на 50 сценаріях із нульовою похибкою, що виключає помилки реалізації формули. По-друге,

модуль оцінки вуглеводів перевірено на репрезентативній вибірці страв типового українського раціону з еталонними значеннями з офіційних таблиць харчової цінності отримана точність 82,0% є обґрунтованою статистичною оцінкою поведінки моделі на реальних даних. По-третє, критичні клінічні стани обробляються виключно детермінованими засобами: мовна модель у цих випадках не викликається, що унеможлиблює будь-яку небезпечну генерацію незалежно від поведінки моделі.

Поєднання цих трьох рівнів перевірки забезпечує системі властивість відмовостійкості. Навіть якщо один із компонентів поводить ся непередбачувано, інші рівні захисту залишаються активними та гарантують безпечний результат для пацієнта. Це особливо важливо в медичних застосунках, де ціна помилки вимірюється не якістю сервісу, а здоров'ям людини.

3 КОМП'ЮТЕРНА МОДЕЛЬ ІНТЕЛЕКТУАЛЬНОГО ЗАСТОСУНКУ

3.1 Обґрунтування вибору середовища програмної реалізації

У рамках кваліфікаційної роботи був розроблений інтелектуальний застосунок для розрахунку дози інсуліну з функціями прогнозування та формування рекомендацій. Для реалізації серверної частини обрано середовище Spring Boot 4.0.5 з мовою програмування Kotlin 2.2.21. Це обумовлено тим, що Spring Boot засновано на високопродуктивній платформі Java, яка забезпечує широкі можливості для побудови надійних серверних застосунків.

Платформа Java надає розробникам потужні засоби для роботи з мережевими з'єднаннями, багатопотоковістю та засоби управління пам'яттю незалежно від операційної системи. Однак пряме використання стандартних бібліотек Java обтяжує програмістів великою кількістю допоміжних дій, зокрема, налаштуванням HTTP-сервера (Hypertext Transfer Protocol), управлінням залежностями та конфігурацією компонентів, що відволікає розробника від основного завдання, а саме реалізації прикладної логіки застосунку.

Spring Boot бере на себе всю допоміжну роботу, звільняючи розробників від написання шаблонного коду з налаштування серверного оточення. Це не означає, що пряме звернення до стандартних бібліотек Java забороняється. При необхідності їх завжди можна використовувати. Однак використання типових серверних функцій компонентами Spring є більш перспективною методикою створення серверних застосунків.

Spring Boot реалізує функції серверної платформи на різних рівнях. Найбільш важливим є спосіб, за допомогою якого компоненти фреймворку обробляють HTTP-запити. При прямому використанні сокетів Java необхідно самостійно керувати з'єднаннями, потоками виконання та серіалізацією

даних. Після завершення обробки запиту розробник зобов'язаний коректно звільнити ресурси та обробити всі можливі виняткові ситуації.

Замість того щоб змушувати працювати з мережевою взаємодією на такому рівні деталізації, Spring Boot надає простий і завершений інтерфейс за допомогою анотацій та компонентної моделі. Фреймворк автоматично керує життєвим циклом компонентів, налаштовує вбудований веб-сервер та забезпечує коректне звільнення ресурсів після завершення обробки запиту. За аналогією з низькорівневими мережевими функціями Java, компоненти Spring мають вкладені властивості, що представляють характеристики обробки запитів, перевірки даних та управління транзакціями бази даних.

Мова програмування Kotlin обрана як основна для реалізації серверної логіки завдяки лаконічності синтаксису та вбудованому захисту від пустих значень на рівні системи типів. Ця властивість є особливо важливою в контексті медичного застосунку, де некоректні вхідні дані можуть призвести до помилкових розрахунків дози інсуліну з безпосередніми наслідками для здоров'я пацієнта.

Для забезпечення безпеки доступу до API (Application Programming Interface) застосовано Spring Security у поєднанні з механізмом JWT (JSON Web Token). Така комбінація дозволяє реалізувати автентифікацію без збереження стану сесії на сервері, що є критично важливим для медичного застосунку з вимогами до масштабованості та конфіденційності даних пацієнтів. Кожен запит до захищених точок входу супроводжується підписаним токеном доступу, а для поновлення сесії без повторної автентифікації використовується окремий токен оновлення. Управління схемою бази даних реалізовано через Flyway, що забезпечує контрольовану та відтворювану міграцію структури даних між версіями застосунку.

Як основну базу даних обрано PostgreSQL 15 з розширенням pgvector. Стандартні реляційні можливості PostgreSQL забезпечують надійне зберігання клінічних даних пацієнтів, історії вимірювань глюкози та журналів ін'єкцій інсуліну. PostgreSQL обрано над альтернативами з кількох

причин: на відміну від MySQL, він підтримує складні типи даних, повноцінні транзакції та розширення; на відміну від Oracle та MS SQL він є повністю безкоштовним без ліцензійних обмежень. Розширення pgvector додає підтримку векторних типів даних та операцій пошуку за косинусною схожістю, що є необхідною умовою для реалізації архітектури RAG. Саме через pgvector зберігаються векторні представлення фрагментів медичних протоколів, що дозволяє системі семантично знаходити релевантні клінічні рекомендації під час формування відповідей мовною моделлю. Вибір єдиної бази даних для реляційних та векторних даних спрощує архітектуру системи та усуває необхідність синхронізації між окремими сховищами.

Для реалізації критично важливих медичних розрахунків створено бібліотеку мовою програмування C, яка компілюється у динамічно підключувану бібліотеку libdiabetes.so для платформи Linux та diabetes.dll для Windows. Вибір мови C для цього компоненту обумовлений детермінованим управлінням пам'яттю та передбачуваним часом виконання без накладних витрат середовища виконання JVM (Java Virtual Machine). Інтеграція бібліотеки з серверною частиною реалізована через JNA (Java Native Access), що дозволяє викликати функції безпосередньо з Kotlin-коду без написання обгортки.

Використання нативних бібліотек мовою C/C++ є обґрунтованим підходом для реалізації алгоритмів на мобільних пристроях, де швидкодія є критичним фактором [20]. Такий підхід дає змогу безпосередньо взаємодіяти з апаратними ресурсами пристрою та ефективніше використовувати процесорний час і пам'ять порівняно з високорівневими мовами програмування. Крім того, нативний код забезпечує нижчі накладні витрати під час виконання обчислювально складних операцій, що особливо важливо для задач обробки сигналів, комп'ютерного зору, машинного навчання та роботи з великими обсягами даних у реальному часі. Використання бібліотек C/C++ також сприяє повторному використанню коду на різних платформах і

дозволяє інтегрувати вже існуючі високопродуктивні програмні рішення без необхідності їх повторної реалізації.

Для реалізації модуля генерації рекомендацій обрано Gemini API від Google, зокрема модель Gemini 3.5 Flash для формування текстових рекомендацій та Gemini Embedding 2 для побудови векторних представлень текстових фрагментів. Порівняльний аналіз доступних мовних моделей, проведений у розділі 2.3, показав, що Gemini 3.5 Flash є оптимальним вибором для медичного застосунку з частими запитами завдяки поєднанню контекстного вікна до 1 мільйону токенів, швидкості генерації 120-170 токенів за секунду та найнижчої вартості серед розглянутих альтернатив. Використання єдиного постачальника для генерації тексту та побудови ембедингів спрощує управління залежностями та забезпечує семантичну узгодженість векторного простору, в якому зберігаються фрагменти медичних протоколів.

Для реалізації клієнтської частини обрано фреймворк Vue.js 3 у поєднанні з фреймворком Ionic 8 та TypeScript. Фреймворк Vue.js надає компонентну архітектуру з підтримкою реактивного оновлення інтерфейсу, що дозволяє відображати зміни глікемічних показників у реальному часі без перезавантаження сторінки. Фреймворк Ionic забезпечує набір компонентів користувацького інтерфейсу, адаптованих під мобільні пристрої, що дозволяє створити єдиний інтерфейс для веб- та Android-платформ. Для візуалізації динаміки рівня глюкози у вигляді інтерактивних графіків використано бібліотеку Chart.js у поєднанні з адаптером vue-chartjs, що надає декларативний інтерфейс для побудови діаграм у межах компонентної моделі Vue. Для забезпечення багатомовності інтерфейсу інтегровано бібліотеку Vue i18n, що дозволяє перемикати мову застосунку без перезавантаження. Розгортання застосунку на платформі Android реалізовано через Capacitor 8, який упаковує веб-застосунок як APK-файл без необхідності написання окремого коду.

Інфраструктура розгортання побудована на основі Docker та Docker Compose. Контейнеризація забезпечує відтворюваність середовища виконання на різних платформах та спрощує розгортання системи, яка складається з кількох взаємозалежних компонентів: серверного застосунку, бази даних PostgreSQL з розширенням pgvector та бібліотеки. Docker Compose описує конфігурацію всіх сервісів в єдиному файлі та автоматично керує мережевою взаємодією між контейнерами, що усуває проблему відмінності середовищ розробки та виробничого розгортання. Збірка серверного застосунку виконується за допомогою Gradle, а керування залежностями клієнтської частини здійснюється через NPM у поєднанні зі збирачем Vite, який забезпечує швидку інкрементальну збірку під час розробки.

3.2 Програмна реалізація

3.2.1 Архітектура системи

Розроблена система реалізована за класичною клієнт-серверною архітектурою і складається з чотирьох основних компонентів: клієнтського застосунку, серверної частини, бази даних та зовнішнього сервісу мовної моделі. Взаємодія між компонентами організована через REST API з використанням протоколу HTTP.

Архітектура з чітким розмежуванням відповідальності між компонентами системи є ефективним підходом при побудові інтелектуальних систем на основі нейронних мереж, що підтверджується досвідом розробки реальних ШІ-застосунків [21].

Клієнтський застосунок, реалізований на основі Vue.js 3 та Ionic 8, виконується у браузері або на Android-пристрої та взаємодіє з сервером виключно через HTTP-запити. Усі запити в застосунку супроводжуються токеном стандарту JWT, який передається у заголовок Authorization. Для

автоматичного оновлення прострочених токенів реалізовано механізм перехоплення відповідей з кодом 401. Клієнт надсилає запит до серверу з токеном оновлення та отримує нову пару токенів без повторної автентифікації користувача. При цьому одночасні запити на оновлення блокуються єдиним прапорцем «isRefreshing», що виключає дублювання операції.

Серверна частина побудована на основі Spring Boot і приймає запити від клієнта через REST API. Усі маршрути серверного застосунку вимагають активного JWT-токена, перевірка якого виконується у фільтрі до передачі запиту до контролера. Серверний застосунок використовує stateless-режим, тобто сесійний стан на сервері не зберігається. Сервер взаємодіє з базою даних для збереження та отримання клінічних даних пацієнта, а для виконання критичних медичних розрахунків звертається до C-бібліотеки через JNA.

Для формування рекомендацій та оцінки вуглеводів сервер надсилає запити до зовнішнього Gemini API. База даних PostgreSQL 15 з розширенням pgvector зберігає два типи даних: реляційні таблиці з клінічними даними пацієнтів та векторні представлення фрагментів медичних протоколів у таблиці «document_chunk», які використовуються для семантичного пошуку в рамках RAG-архітектури.

C-бібліотека компілюється окремим Docker-контейнером під час запуску системи та монтується у файлову систему серверного контейнера. Це забезпечує детерміноване виконання розрахунків без накладних витрат JVM. Розгортання компонентів описано у єдиному файлі «docker-compose.yml» та виконується послідовно з урахуванням залежностей: спочатку запускається база даних і очікує стану «healthy», потім збирається бібліотека, після чого стартує серверний застосунок, і лише потім клієнтська частина. Таким чином сукупність описаних компонентів та зв'язки між ними формують послідовну взаємодію системи (рис. 3.1).

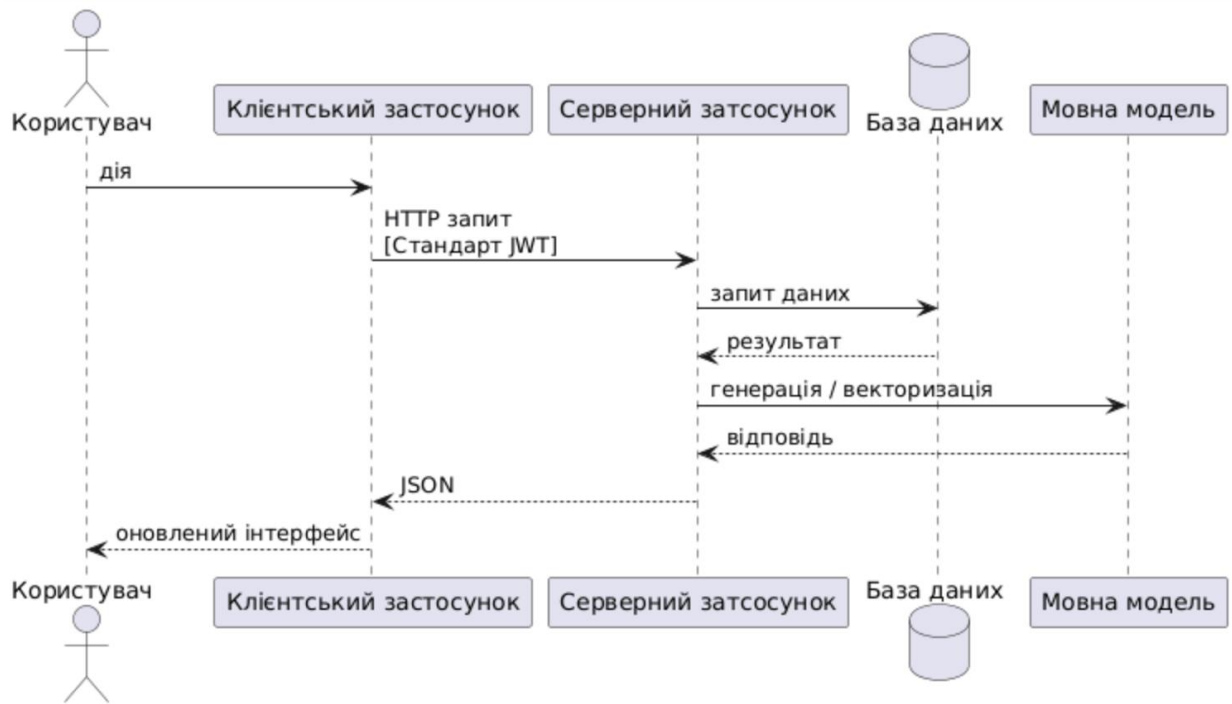


Рисунок 3.1 – Діаграма взаємодії системи

Розроблена архітектура забезпечує чітке розмежування відповідальності між компонентами системи: клієнтський застосунок відповідає за відображення даних та взаємодію з користувачем, серверний застосунок відповідає за бізнес-логіку, безпеку та оркестрацію запитів, база даних відповідає за надійне зберігання як реляційних, так і векторних даних. Використання контейнеризації через Docker гарантує відтворюваність середовища та коректний порядок ініціалізації залежностей, що в сукупності робить систему керованою, масштабованою та придатною для розгортання в реальних умовах.

3.2.2 Розробка серверного застосунку

Серверна частина реалізована на мові Kotlin і організована за принципом пошарової та пакетної архітектури. Кореневий пакет `ua.nure.smartinsulinbackend` містить підпакети `controller`, `service`, `repository`, `entity`, `dto` та `config`, що забезпечує чітке розмежування відповідальності між

рівнями системи та спрощує її супроводження. Взаємодія між клієнтом і сервером відбувається через REST API з базовим шляхом /api. Маршрути в застосунку визначаються класами-контролерами. В даному застосунку реалізовано такі класи-контролери: AuthController, BolusController, GlucoseReadingController, ForecastController, MealRecordController, InsulinDoseController та UserProfileController.

Клас AuthController надає чотири маршрути автентифікації: реєстрацію, вхід, оновлення токена та вихід із системи. Клас BolusController забезпечує основну медичну функціональність: розрахунок болюсної дози за алгоритмом формули 2.1, оцінку вуглеводів за текстовим описом страви та генерацію персоналізованої рекомендації.

Клас GlucoseReadingController реалізує CRUD-операції для показників глікемії та розрахунок глікованого гемоглобіну HbA1c через C-бібліотеку. ForecastController приймає запит на прогнозування та повертає очікуваний рівень глюкози з діапазоном невизначеності та прапорцями ризику «HYPO_RISK» і «HYPER_RISK». Класи MealRecordController та InsulinDoseController забезпечують управління записами про прийоми їжі та введення інсуліну відповідно. UserProfileController надає доступ до профілю пацієнта та ручний запуск перерахунку адаптивних коефіцієнтів ICR та ISF.

Повний перелік маршрутів із зазначенням HTTP-методу, шляху та вимог до авторизації наведено у таблиці 3.1.

Таблиця 3.1 – Список всіх маршрутів застосунку

Метод	Шлях	Опис	Авторизація
1	2	3	4
POST	/api/auth/register	Реєстрація нового користувача	Ні
POST	/api/auth/login	Вхід, повертає пару JWT-токенів	Ні
POST	/api/auth/refresh	Оновлення токена через refresh-токен	Ні
POST	/api/auth/logout	Вихід, інвалідація refresh-токена	Так

Продовження таблиці 3.1

1	2	3	4
POST	/api/bolus/recommendation	Генерація клінічної рекомендації через LLM з RAG	Так
GET	/api/glucose	Отримання списку вимірювань глюкози	Так
POST	/api/glucose	Додавання нового вимірювання	Так
PUT	/api/glucose/{id}	Редагування вимірювання	Так
DELETE	/api/glucose/{id}	Видалення вимірювання	Так
GET	/api/glucose/hbA1c	Розрахунок HbA1c за останні 90 днів через C-бібліотеку	Так
GET	/api/forecast	Прогноз рівня глюкози на горизонт до 180 хвилин	Так
GET	/api/meals	Отримання списку записів про прийоми їжі	Так
POST	/api/meals	Додавання запису про прийом їжі	Так
PUT	/api/meals/{id}	Редагування запису	Так
DELETE	/api/meals/{id}	Видалення запису	Так
GET	/api/insulin-doses	Отримання списку введених доз інсуліну	Так
POST	/api/insulin-doses	Додавання запису про введення дози	Так
PUT	/api/insulin-doses/{id}	Редагування запису	Так
DELETE	/api/insulin-doses/{id}	Видалення запису	Так
GET	/api/profile	Отримання профілю пацієнта та клінічних параметрів	Так
PUT	/api/profile	Оновлення профілю та параметрів	Так
POST	/api/profile/recalculate-coefficients	Ручний запуск перерахунку коефіцієнтів ICR та ISF	Так

Для захисту API застосовано stateless JWT-автентифікацію на основі пари tokenів – access-токена та refresh-токена. Access-токен підписується алгоритмом HMAC-SHA256 і передається клієнтом при кожному запиті у HTTP-заголовок Authorization з префіксом Bearer. Сервер, отримавши запит, перевіряє підпис токена за допомогою секретного ключа та перевіряє термін його дії. Якщо токен прострочений або підпис не збігається, запит відхиляється з поверненням HTTP-статусу 401 Unauthorized. Refresh-токен має значно довший термін дії і призначений виключно для отримання нової пари tokenів після закінчення терміну дії access-токена. У базі даних він зберігається не у відкритому вигляді, а у формі SHA-256 хешу, що унеможливорює його відновлення у разі витоку даних зі сховища.

Обробку tokenів на рівні HTTP-запитів реалізує клас JwtAuthenticationFilter, який впроваджується у ланцюжок фільтрів Spring Security перед стандартним фільтром UsernamePasswordAuthenticationFilter. Це гарантує, що кожен вхідний запит проходить JWT-верифікацію ще до того, як Spring Security застосує власну логіку автентифікації. Фільтр витягує токен із заголовка запиту, виконує перевірку його підпису та терміну дії, і у разі успіху встановлює контекст автентифікації для поточного запиту, дозволяючи йому пройти далі по ланцюжку.

Загальні правила розмежування доступу визначені у методі securityFilterChain класу SecurityConfig. У цій конфігурації CSRF-захист відключено як надлишковий, бо він актуальний для додатків із сесійною автентифікацією, тоді як stateless API, що не використовує cookies для передачі облікових даних, не схильний до відповідного типу атак. Публічний доступ без автентифікації відкрито лише для маршрутів /api/auth/**, що охоплюють реєстрацію, вхід та оновлення tokenів, а також для службового маршруту /error. Маршрут /error дозволяє повертати помилки на клієнт. До помилок можуть відносити як системні помилки, так і програмні (неправильні значення, тощо). Найпоширеніше використання даного маршруту виникає при розробці – помилка «404 не знайдено». Усі інші

маршрути вимагають наявності актуального access-токена у заголовку запиту. Для реалізації описаної схеми режим управління сесіями встановлено у значення «STATELESS», тобто сервер не створює і не зберігає HTTP-сесії між запитами, а кожен запит є повністю самодостатнім з точки зору автентифікації, що цілком відповідає принципам REST-архітектури.

Модель даних системи складається з восьми JPA-сутностей (Java Persistence API), пов'язаних відносинами через зовнішні ключі до сутності User. Сутність GlucoseReading зберігає показник рівня глюкози в крові у ммоль/л разом із міткою часу вимірювання та ідентифікатором джерела. Сутність InsulinDose фіксує введену дозу інсуліну, її тип та час введення і використовується для розрахунку активного інсуліну IOB.

Сутність MealRecord містить текстовий опис прийому їжі та оцінену кількість вуглеводів у грамах. Сутність UserProfile зберігає індивідуальні параметри пацієнта: коефіцієнти ICR та ISF, цільовий рівень глюкози та тривалість дії інсуліну. Сутність DocumentChunk призначена для реалізації RAG-архітектури та містить текстовий фрагмент медичного документа й векторне представлення у форматі vector(3072), отримане від моделі Gemini Embedding 2, а розширення pgvector забезпечує індексування та семантичний пошук за косинусною подібністю.

Управління схемою бази даних реалізовано через інструмент Flyway, який автоматично застосовує версійовані SQL-скрипти при кожному старті застосунку. Скрипти розташовані у директорії міграцій db/migration і іменуються за конвенцією «V{номер}__{опис}.sql». Ланцюжок міграцій налічує десять версій.

Міграція V8 активує розширення pgvector та створює таблицю document_chunks з HNSW-індексом (Hierarchical Navigable Small World) для пошуку найближчих сусідів за косинусною відстанню. Ця таблиця є одною з головних в системі і фактично ключова для побудови RAG. Структура сутності зображено в таблиці 3.2.

Таблиця 3.2 – Структура сутності document_chunks в базі даних

Поле	Тип	Опис
id	bigserial	Первинний ключ
source	varchar(255)	Джерело доступу
chunk_index	int	Номер фрагменту
content	text	Текстовий вміст фрагменту
embedding	vector(3072)	Векторне представлення фрагменту

Поле embedding типу vector(3072) зберігає числовий вектор, отриманий від моделі Gemini Embedding 2. Розмірність 3072 відповідає вихідному простору цієї моделі та забезпечує достатню точність семантичного представлення медичних текстів. Індекс HNSW на цьому полі дозволяє виконувати наближений пошук найближчих сусідів за мілісекунди навіть при великій кількості збережених фрагментів, що є критично важливим для часу відповіді RAG-модуля в реальному часі.

Таким чином, за допомогою JPA та Flyway, серверний застосунок формує консистентну модель даних, яка задовольняє правила реляційних баз даних (рис. 3.2).

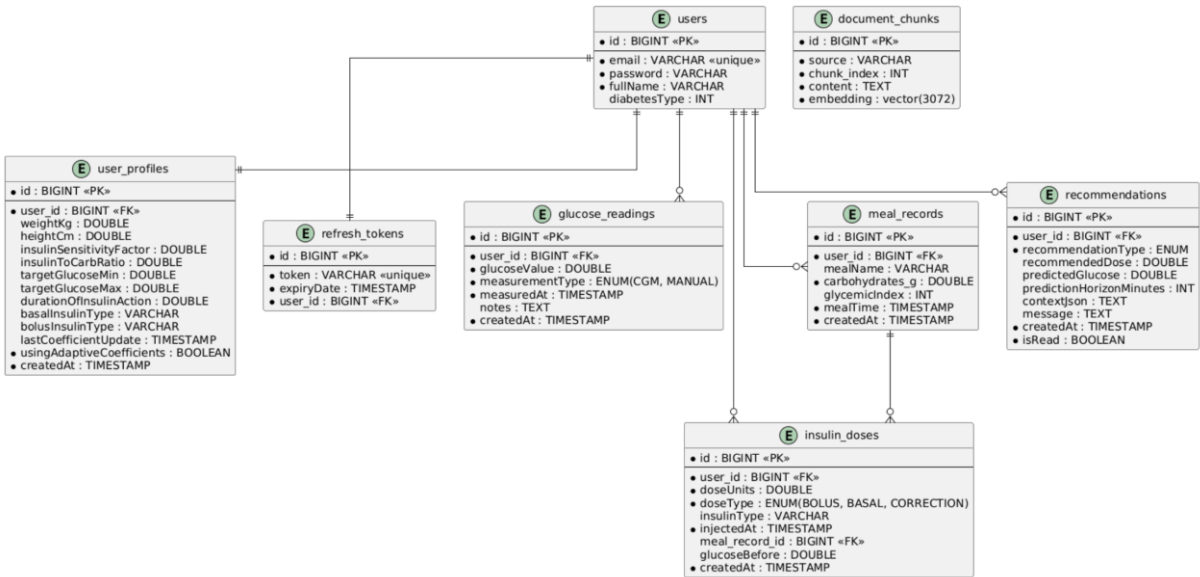


Рисунок 3.2 – Модель даних застосунку

Модель даних спроектована за принципом єдиної точки відліку, тобто всі клінічні сутності пов'язані з таблицею User через зовнішні ключі, що спрощує контроль доступу та забезпечує ізоляцію даних між пацієнтами. Таблиця document_chunks є винятком з цієї схеми: вона не прив'язана до конкретного користувача, оскільки зберігає загальні медичні протоколи, доступні для всіх запитів системи. Така архітектура чітко розмежовує персональні клінічні дані та базу знань, на якій працює RAG.

Таким чином, серверний застосунок реалізує повний набір функціональності, необхідної для підтримки клінічних процесів: від автентифікації та захисту API до медичних розрахунків, прогнозування та генерації рекомендацій. Пошарова пакетна архітектура на основі Kotlin і Spring Boot забезпечує чітке розмежування відповідальності між компонентами, а stateless JWT-автентифікація відповідає принципам безпечного REST-дизайну. Версійоване управління схемою через Flyway та використання pgvector для семантичного пошуку формують надійну та розширювану основу для зберігання як реляційних клінічних даних, так і векторних представлень медичних протоколів. Розподіл на контролери, сервіси та репозиторії дозволяє незалежно тестувати та розширювати кожен шар без змін в інших частинах системи. Така організація коду знижує зв'язність між компонентами та спрощує подальший супровід застосунку в умовах зростання функціональних вимог.

3.2.3 Розробка модулів обчислень та штучного інтелекту

Обчислювальна логіка системи реалізована у вигляді п'яти сервісних класів та окремої C-бібліотеки libdiabetes. Нативна бібліотека є спільною математичною основою обчислювального рівня. Вона задіяна у трьох сервісах – розрахунку болюсної дози, прогнозуванні глюкози та обчисленні HbA1c.

Бібліотека `libdiabetes` реалізована мовою C та компілюється у форматі спільного об'єкта (`libdiabetes.so`) у рамках окремого Docker. Скомпільований компонент монтується до контейнера серверної частини через спільний том та підключається через механізм JNA. Інтерфейс `HbA1cLibrary` оголошує сигнатури нативних функцій, а JNA автоматично здійснює перетворення типів між Kotlin та C при кожному виклику. Бібліотека реалізує три групи функцій: розрахунок глікованого гемоглобіну HbA1c, модель лінійного розпаду активного інсуліну IOB та функції прогнозування.

Публічний інтерфейс бібліотеки визначено у файлі заголовків `diabetes.h`. Для забезпечення крос-платформної сумісності введено макрос `EXPORT`, який визначається як `__declspec(dllexport)` при компіляції під Windows та як `__attribute__((visibility("default")))` під Linux. Це дозволяє використовувати бібліотеку як у вигляді `libdiabetes.dll`, так і у вигляді `libdiabetes.so` без змін у вихідному коді. Заголовковий файл містить захисний блок `#ifdef __cplusplus`, що дозволяє підключати бібліотеку з C++ без калічення символів. Функції в бібліотеці умовно можна поділити за призначенням: прогнозування, HbA1c, інсулін, статистика. Заголовковий файл збирає всі визначення функцій в бібліотеці, тому представлення бібліотеки можна зобразити у вигляді діаграми пакетів (рис. 3.3).

Модель розпаду активного інсуліну реалізована у вигляді лінійної функції `calculate_iob`, яка приймає три параметри: початкову дозу інсуліну `dose`, час після ін'єкції `time_since_inj` та тривалість дії інсуліну `dia`. Функція реалізує лінійну модель розпаду та повертає залишкову кількість активного інсуліну в одиницях. Для крайніх випадків передбачено граничні умови: якщо час після ін'єкції менший або рівний нулю – повертається повна доза, якщо він перевищує або дорівнює тривалості дії інсуліну. Функція повертає нуль, що означає повне виведення інсуліну з організму.

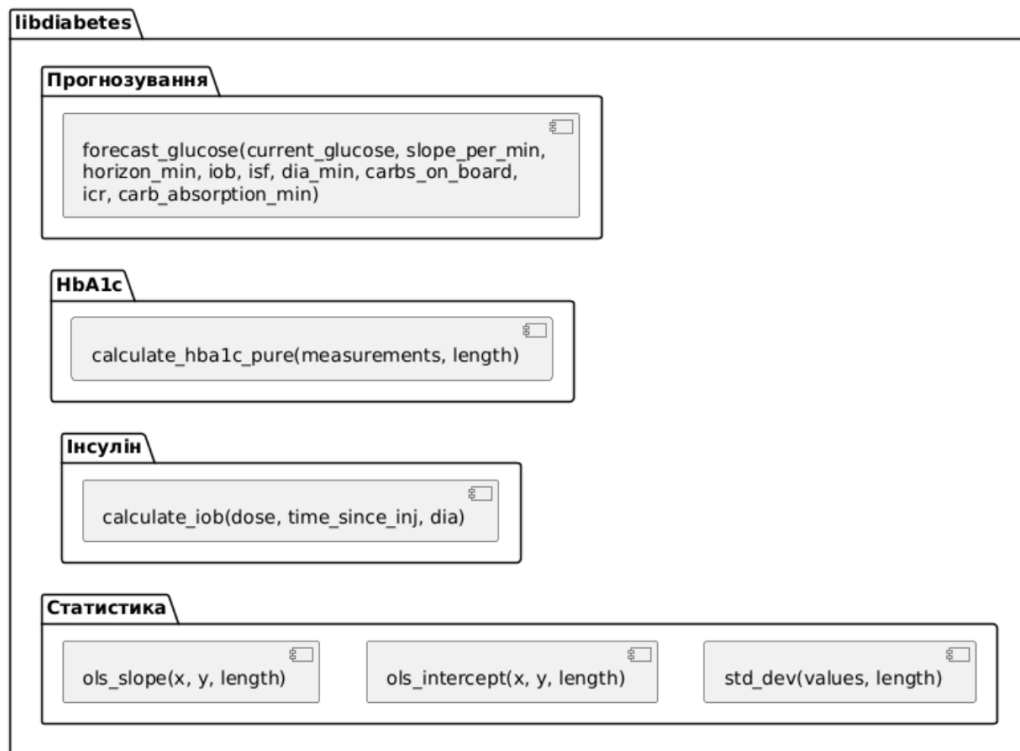


Рисунок 3.3 – Структура C бібліотеки

Функція прогнозування `forecast_glucose` має три складові: загасаючий лінійний тренд, знижувальний ефект активного інсуліну та підвищувальний ефект вуглеводів що продовжують засвоюватись. Загасання тренду реалізовано через експоненційний множник, що відображає зниження точності лінійної екстраполяції зі збільшенням горизонту прогнозу.

Алгоритм функції виконується у чотири етапи.

На першому етапі перевіряється гранична умова: якщо горизонт прогнозу менший або рівний нулю, функція негайно повертає поточний рівень глюкози без обчислень.

На другому етапі обчислюється загасаючий тренд. Коефіцієнт загасання визначається експоненційно залежно від горизонту прогнозу: чим більший горизонт, тим менший вплив тренду на результат. Константа 120 хвилин відповідає характерному часу зниження точності лінійної екстраполяції. Тренд дорівнює добутку поточного нахилу глюкози, горизонту прогнозу та коефіцієнта загасання. Завдяки цьому для коротких

горизонтів (15-30 хвилин) тренд майже не загасає, тоді як для довгих проміжках (90-120 хвилин) його вплив суттєво зменшується.

На третьому етапі обчислюється знижувальний ефект активного інсуліну. Якщо всі три параметри – активний інсулін IOB, коефіцієнт чутливості ISF та тривалість дії інсуліну `dia_min` є додатними, визначається частка горизонту від загальної тривалості дії інсуліну, обмежена константним значенням 1,0. Ця частка, помножена на IOB та ISF, дає очікуване зниження рівня глюкози від залишкового інсуліну.

На четвертому етапі обчислюється підвищувальний ефект вуглеводів. Кількість засвоєних вуглеводів за горизонт прогнозу визначається пропорційно до часу засвоєння `carb_absorption_min`, також обмежена зверху. Підвищення глюкози від вуглеводів розраховується через співвідношення ISF до ICR, що відображає індивідуальну реакцію пацієнта на вуглеводи.

Фінальний прогноз формується як сума поточного рівня глюкози, тренду, знижувального ефекту інсуліну та підвищувального ефекту вуглеводів. Для уникнення фізично неможливих значень результат обмежується нижньою межею 1,0 ммоль/л.

Підключення бібліотеки до серверного застосунку реалізовано у конфігураційному класі `NativeLibraryConfig`. Метод `Native.load(«diabetes», HbA1cLibrary::class.java)` завантажує бібліотеку за іменем `diabetes`. JNA автоматично додає платформозалежний префікс і розширення (`libdiabetes.so` на Linux, `libdiabetes.dll` на Windows) та шукає файл у системних шляхах бібліотек. Завантажений екземпляр реєструється як сервіс і впроваджується через конструктор у всі сервіси, що його потребують. Бібліотека підключається в інші класи за допомогою механізму `Dependency Injection`, який гарантує що буде створено єдиний екземпляр коду. Клас позначено анотацією `Configuration`, а метод `Bean` інструктує Spring створити єдиний екземпляр бібліотеки та керувати його життєвим циклом у контексті застосунку.

Клас `BolusCalculationService` реалізує формулу розрахунку болюсної дози. Сервіс отримує коефіцієнти ICR та ISF з профілю пацієнта, обчислює дозу на вуглеводи, корекційну дозу та значення IOB. Значення IOB розраховується зовнішньою функцією `calculate_iob` для кожної дози у межах вікна дії інсуліну. Клас `AdaptiveCoefficientService` щотижня перераховує персональні коефіцієнти ICR та ISF на основі тридцятиденної історії пацієнта.

Сервісний рівень системи побудований таким чином, щоб забезпечити розділення логіки обробки даних між незалежними компонентами. Кожен сервіс відповідає за окремий функціональний напрям і реалізує завершений набір операцій, необхідних для виконання відповідних бізнес-процесів. Такий підхід зменшує зв'язність між модулями, спрощує супровід програмного забезпечення та підвищує його масштабованість. Для обміну даними між сервісами використовуються стандартизовані структури запитів і відповідей, що забезпечує узгодженість роботи окремих компонентів системи. Крім того, ізоляція функціональності в межах окремих сервісів дозволяє незалежно вдосконалювати алгоритми розрахунків, механізми персоналізації та інтеграцію із зовнішніми сервісами без необхідності внесення суттєвих змін до інших частин програмного комплексу.

Сервіс `GeminiService` забезпечує дві функції: оцінку вмісту вуглеводів за текстовим описом страви та генерацію клінічних рекомендацій. Перед зверненням до мовної моделі Gemini 3.5 Flash виконується перевірка безпеки.

Генерація клінічної рекомендації реалізована у методі `generateRecommendation` класу `GeminiService` і складається з чотирьох послідовних етапів: перевірки безпеки, формування RAG-запиту, побудови інструкцій моделі та перевірки відповіді. Послідовність алгоритму можна зобразити у вигляді блок-схеми (рис. 3.4).



Рисунок 3.4 – Алгоритм генерації рекомендацій у вигляді блок-схеми

При глюкозі нижче 3,9 ммоль/л повертається фіксоване попередження без виклику моделі. Також в коді перевіряються інші небезпечні сценарії, такі як початкова гіперглікемія, велика розрахована доза болюсу. Визначення прапорів безпеки реалізовано у вигляді переліку правил, що дозволяє легко додати нові правила, або редагувати існуючі. Функція повертає набір прапорів, що означають набір обмежень. Список прапорів, умов появи і описів зображено в таблиці 3.3.

Таблиця 3.3 – Список прапорів-обмежень

Прапор	Умова	Опис
HYPOGLYCEMIA	глюкоза < 3,9 ммоль/л	Небезпечно низький рівень глюкози
HIGH GLUCOSE	глюкоза > 10,0 ммоль/л	Початкова гіперглікемія
HIGH IOB	IOB > 2,5 од.	Високий активний інсулін
LARGE_DOSE	доза > 15,0 од.	Велика розрахована доза болюсу

Метод `buildRagQuery` формує текстовий запит для семантичного пошуку на основі виявлених прапорців, після чого `retrieveProtocolContext` отримує релевантні фрагменти медичних протоколів, які формують контекст для системної інструкції моделі. Метод `retrieveProtocolContext` виконує семантичний пошук релевантних фрагментів медичних протоколів у три кроки.

На першому кроці текстовий запит передається до методу `embed` класу `EmbeddingService`, який надсилає запит до Gemini Embedding 2 API та повертає числовий вектор розмірністю 3072.

На другому кроці отриманий вектор передається до методу `findTopSimilar` класу `DocumentChunkRepository`, який виконує SQL-запит до таблиці `document_chunks` з використанням оператора косинусної подібності `pgvector` та повертає п'ять найближчих фрагментів, впорядкованих за релевантністю. Тобто чим менший кут між векторами запиту та фрагменту, тим вищий його рейтинг.

На третьому кроці метод `buildString` формує з отриманих фрагментів єдиний текстовий блок, де кожен фрагмент супроводжується назвою джерела та порядковим номером і обмежується 600 символами для уникнення перевищення контекстного вікна моделі.

Для отримання релевантного контексту з медичних протоколів `GeminiService` координує взаємодію між двома залежними компонентами: `EmbeddingService`, відповідальним за векторизацію запиту, та `DocumentChunkRepository`, який виконує пошук у базі даних (рис. 3.5).

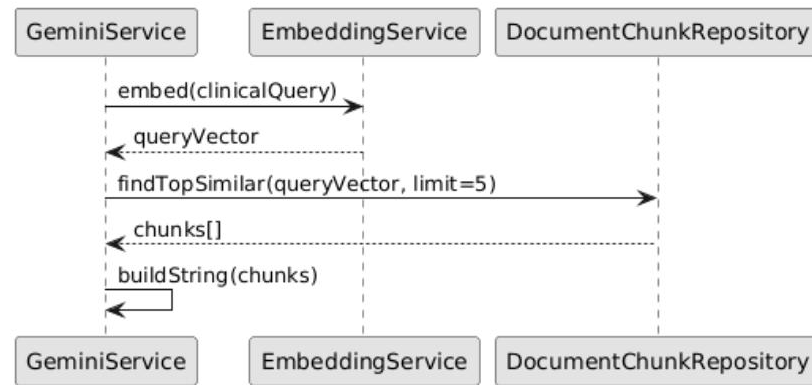


Рисунок 3.5 – Діаграма послідовності отримання контексту з медичних протоколів

При формуванні рекомендації GeminiService знаходить п'ять найближчих фрагментів за косинусною подібністю через pgvector та включає їх до системної інструкції моделі. Далі запит формується з двох частин: системної і клінічної. До системних інструкцій додається контекст отриманих протоколів та дані по прапорах. Клінічні інструкції повністю складаються з математично розрахованих даних, щодо стану пацієнта.

Системні інструкції формуються з двох блоків.

Перший визначає роль моделі: «медичний асистент для пацієнтів з діабетом I типу» та правила відповіді: «українська мова», «зрозуміла для пацієнта без медичної освіти», «не більше чотирьох-п'яти речень», «без пропозиції альтернативної дози та з обов'язковим застереженням щодо заміни консультації лікаря».

Другий блок містить контекст релевантних фрагментів медичних протоколів, отриманих через семантичний пошук. Клінічний запит формується окремо і містить поточний рівень глюкози з трендом, кількість вуглеводів, дозу на вуглеводи, корекційну дозу, активний інсулін, загальну розраховану дозу, заплановану фізичну активність та виявлені прапори безпеки.

Після отримання відповіді від мовної моделі виконуються додаткові перевірки за допомогою механізму Regex. Виконується пошук патернів, які

потенційно можуть нашкодити пацієнту, наприклад «збільшіть дозу», «введіть».

Таким чином, поєднання нативної C-бібліотеки для детермінованих математичних обчислень, адаптивних сервісних класів Kotlin та RAG-збагаченої інтеграції з Gemini формує повноцінний обчислювальний рівень системи, здатний як точно розраховувати дози інсуліну, так і генерувати клінічно обґрунтовані рекомендації з урахуванням актуальних медичних протоколів та індивідуального стану пацієнта.

3.2.4 Розробка клієнтського застосунку

В якості клієнтської частини реалізовуємо застосунок на основі Vue.js 3 з використанням UI-фреймворку Ionic 8. Збірка здійснюється інструментом Vite, адреса серверного API передається через змінну оточення `VITE_API_BASE_URL` на етапі побудови Docker-образу. Застосунок підтримує локалізацію двома мовами – українською та англійською. Локалізація реалізується за допомогою `vue-i18n`. Для розгортання застосунку у вигляді Android-застосунку використовується Capacitor 8, який обгортає застосунок у Android оболонку без змін у вихідному коді.

Точкою входу є файл `main.ts`, який створює екземпляр Vue та підключає три плагіни: `IonicVue` для компонентів інтерфейсу, `vue-router` для маршрутизації та `i18n` для локалізації. Монтування застосунку відкладається до ініціалізації маршрутизатора через `router.isReady()`, що гарантує коректне виконання захисту маршрутів перед першим рендерингом.

Маршрутизація реалізована через `vue-router` з ледачим завантаженням сторінок, яке дозволяє завантажувати лише необхідні ресурси в час виконання коду. Застосунок містить дев'ять маршрутів: сторінки автентифікації (`/login`, `/register`) позначені мета-полем `public`, решта маршрутів захищені. Захист маршрутів реалізовано через глобальний

навігаційний охоронець `beforeEach`. При першому переході ініціалізується стан автентифікації через перевірку наявності збереженого токена; неавтентифікований користувач перенаправляється на автентифікації, а автентифікований при спробі відкрити сторінку входу буде направлений на головну сторінку.

Стан автентифікації зберігається у реактивній змінній `isAuthenticated` типу `Ref<boolean>` із модуля `auth.ts`. Токени зберігаються через плагін `Preferences` бібліотеки `Capacitor`, що на мобільній платформі використовує нативне захищене сховище, а у браузері – `localStorage`. Функція `apiFetch` є єдиною точкою звернення до API: вона автоматично додає заголовок `Authorization: Bearer` до кожного запиту та обробляє відповідь зі статусом 401 шляхом оновлення токена через `/api/auth/refresh`. Прапорець `isRefreshing` запобігає паралельному запуску кількох запитів на оновлення при одночасному отриманні кількох 401-відповідей.

Головна сторінка `DashboardPage` відображає чотири блоки інформації: прогноз глюкози у вигляді лінійного графіка, розрахований `HbA1c` за останні 90 днів, форму додавання нового вимірювання та пагінований список попередніх записів. Графік прогнозу побудовано бібліотекою `Chart.js` через обгортку `vue-chartjs` і відображає прогнозну лінію з кольоровою смугою невизначеності. Значення `HbA1c` забарвлюється залежно від рівня компенсації: зелений колір відповідає значенню нижче 7%, жовтий 7-8%, червоний вище 8%. Такий підхід покращує клієнтський досвід користування застосунком.

Сторінка `BolusPage` реалізує повний сценарій розрахунку болюсної дози. Форма містить поля для введення рівня глюкози, опису страви та кількості вуглеводів, а також блок параметрів фізичної активності з вибором типу, інтенсивності, часу до початку та тривалості. Кнопка AI-оцінки вуглеводів надсилає текстовий опис страви до серверного застосунку і підставляє отриманий результат у поле вуглеводів. Після розрахунку відображається картка з розбивкою дози на складові. Також для покращення

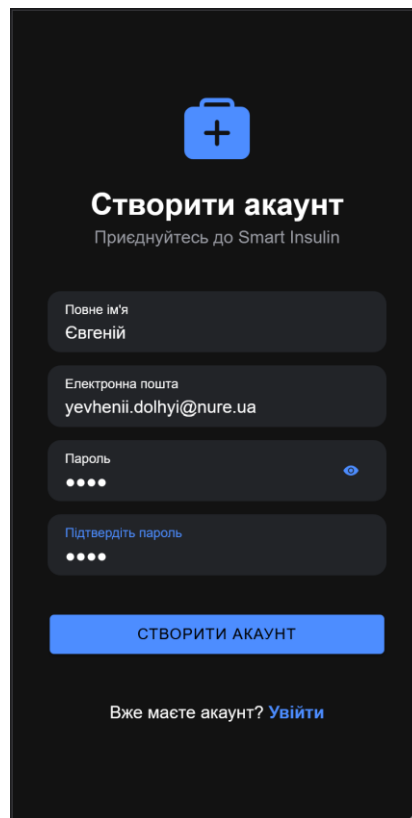
користувачам розуміння стану здоров'я, застосунок відображає прапорці безпеки, повернуті сервером. Прапорці відображаються кольоровими мітками відповідно до їхньої критичності.

Окремою сторінкою застосунку є ProfilePage, що дозволяє пацієнту переглядати та редагувати персональні дані. Користувач може змінити свої антропометричні показники: вага та зріст, що використовуються для контексту при розрахунках, клінічні параметри інсулінотерапії: коефіцієнти ISF та ICR, тривалість дії інсуліну, а також тип базального та болюсного інсуліну. Також користувач може змінити нижню та верхню межі глюкози, що безпосередньо впливають на розрахунок корекційної дози.

Окрім медичних параметрів, сторінка містить блок налаштувань застосунку з перемикачем мови інтерфейсу між українською та англійською, а також теми застосунку. Обраний варіант зберігається локально та застосовується негайно без перезавантаження.

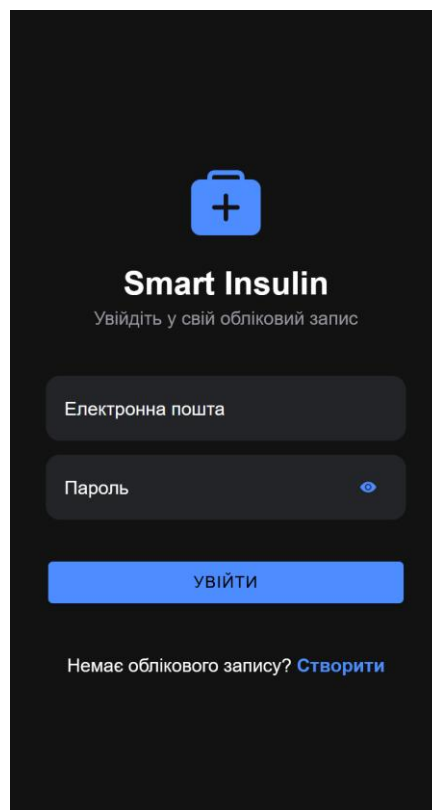
3.3 Інструкція користувача

Для початку роботи із застосунком користувач повинен пройти реєстрацію, вказавши електронну адресу, пароль та повне ім'я (рис. 3.6). Поле пароля підтримує перемикання видимості символів для зручності введення. Після заповнення всіх полів та натискання кнопки створення облікового запису система автоматично виконує вхід і перенаправляє до головної сторінки. У разі помилки (наприклад, якщо електронна адреса вже використовується) під формою відображається відповідне повідомлення. При повторних відвідуваннях достатньо ввести електронну адресу та пароль на сторінці входу (рис. 3.7). При вводі неправильних даних застосунок покаже помилку на формі. Якщо користувач вже автентифікований, спроба відкрити сторінку входу або реєстрації автоматично перенаправляє до головної сторінки.



The registration screen features a blue medical bag icon with a white plus sign at the top. Below it, the title "Створити акаунт" (Create account) is displayed in white, followed by the subtitle "Приєднуйтесь до Smart Insulin" (Join Smart Insulin). The form contains four input fields: "Повне ім'я" (Full name) with the value "Євгеній", "Електронна пошта" (Email) with the value "yevhenii.dolhyi@nure.ua", "Пароль" (Password) with four dots, and "Підтвердіть пароль" (Confirm password) with four dots. A blue "СТВОРИТИ АКАУНТ" (Create account) button is positioned below the fields. At the bottom, a link "Вже маєте акаунт? Увійти" (Already have an account? Log in) is shown in blue.

Рисунок 3.6 – Сторінка реєстрації в системі



The login screen features the same blue medical bag icon with a white plus sign at the top. Below it, the title "Smart Insulin" is displayed in white, followed by the subtitle "Увійдіть у свій обліковий запис" (Log in to your account). The form contains two input fields: "Електронна пошта" (Email) and "Пароль" (Password) with four dots. A blue "УВІЙТИ" (Log in) button is positioned below the fields. At the bottom, a link "Немає облікового запису? Створити" (Don't have an account? Create) is shown in blue.

Рисунок 3.7 – Сторінка входу в систему

Після входу відкривається головна сторінка, що містить чотири інформаційні блоки. У першому блоці відображається прогноз рівня глюкози у вигляді лінійного графіка з кольоровою смугою невизначеності на горизонт до 180 хвилин. Поряд із графіком показується поточний рівень глюкози та вектор тренду – стрілка зі швидкістю зміни у ммоль/л на годину. При наявності ризику гіпоглікемії або гіперглікемії над графіком відображаються відповідні кольорові мітки попередження. У другому блоці показується розрахований рівень HbA1c з кольоровим індикатором компенсації (рис. 3.8). Третій блок використовується для додавання нового вимірювання глюкози користувач вводить значення у ммоль/л, обирає тип вимірювання (ручне або CGM) та натискає кнопку збереження. В четвертому блоці розташовано список всіх вимірювань підтримує посторінкове завантаження (рис. 3.9). Список попередніх вимірювань дозволяє видаляти та редагувати записи. Для того щоб відкрити запис для видалення або редагування потрібно провести пальцем вліво для видалення і вправо для редагування (рис. 3.10).

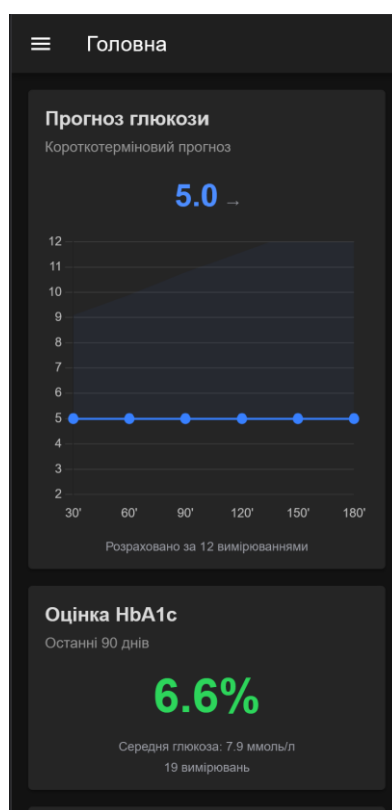


Рисунок 3.8 – Головна сторінка системи (частина 1)

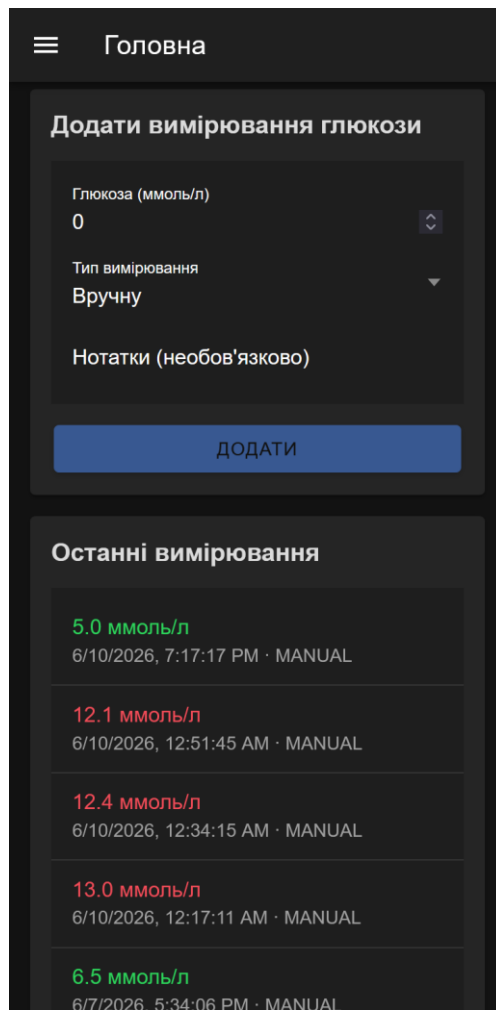


Рисунок 3.9 – Головна сторінка системи (частина 2)

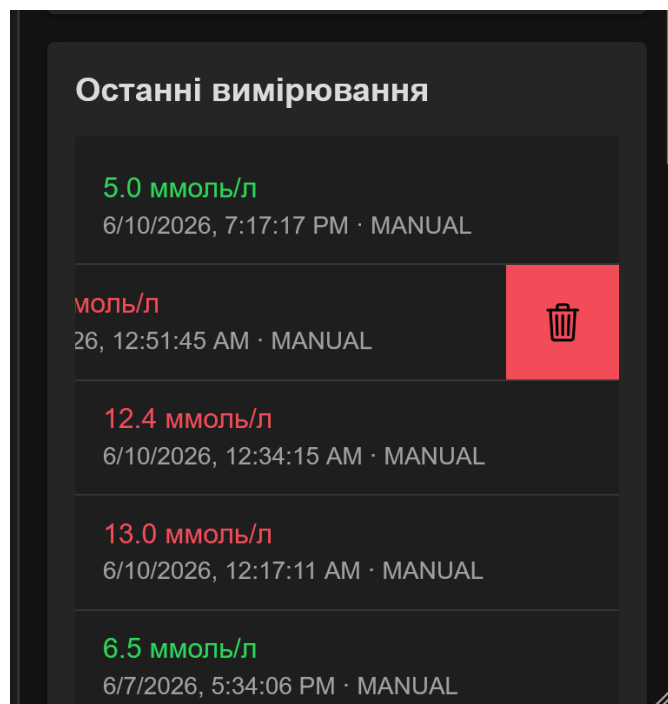


Рисунок 3.10 – Кнопка видалення запису

Для того щоб перейти на іншу сторінку потрібно натиснути на кнопку меню, яка має вигляд трьох горизонтальних прямих в лівому верхньому куті. Після натискання відкриється меню з переліком сторінок (рис. 3.11).

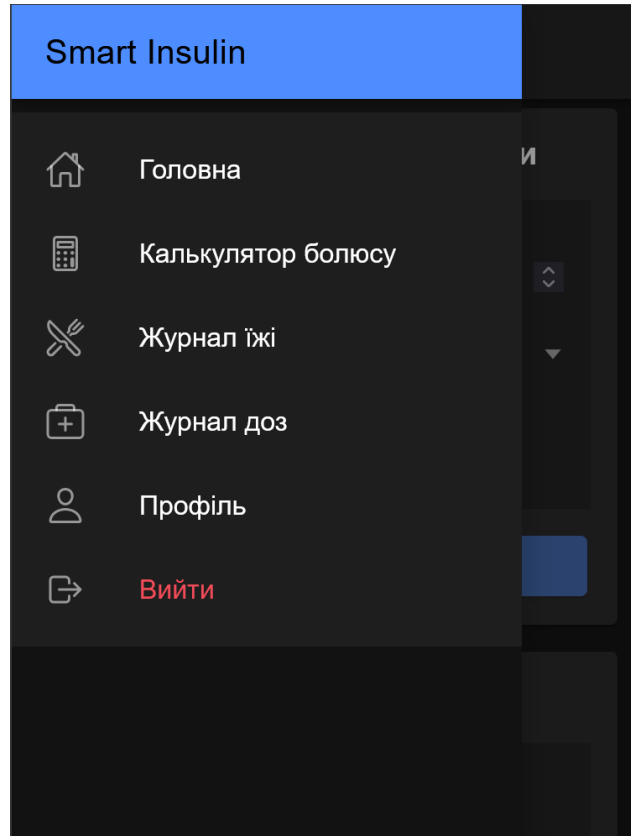


Рисунок 3.11 – Кнопка видалення запису

Для розрахунку болюсної дози користувач повинен обрати в меню пункт з назвою «Калькулятор болюсу». Після натискання застосунок відкриє сторінку з полями вводу і списком останніх ін'єкцій (рис. 3.12). В формі обов'язковими полями є поточний рівень глюкози та кількість вуглеводів у страві. Для автоматичної оцінки вуглеводів користувач може ввести текстовий опис страви у довільній формі і натиснути кнопку «Визначити вуглеводи». Тоді система надсилає опис до мовної моделі та підставить отриману оцінку у поле вуглеводів (рис. 3.13). Глікемічний індекс є необов'язковим параметром. В разі заповнення глікемічного індексу після розрахунку відображається підказка щодо рекомендованого часу введення інсуліну відносно прийому їжі. Якщо заплановано фізичне навантаження,

користувач розгортає відповідний блок та вказує тип активності, інтенсивність, час до початку та тривалість (рис. 3.14). Система автоматично знизить дозу з урахуванням цих параметрів.

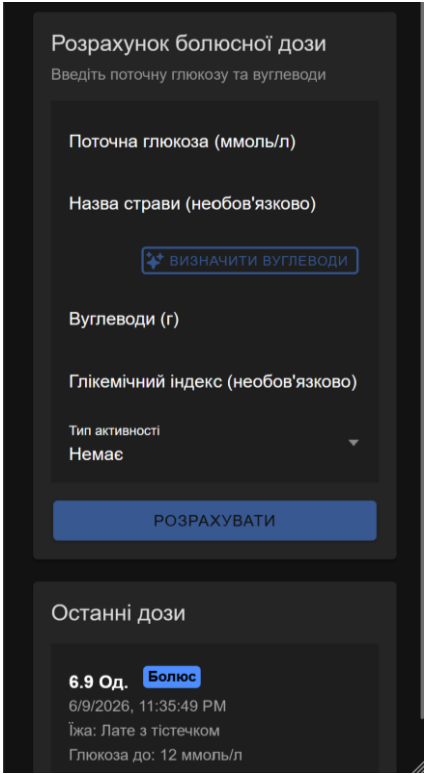


Рисунок 3.12 – Сторінка калькулятора болюсу

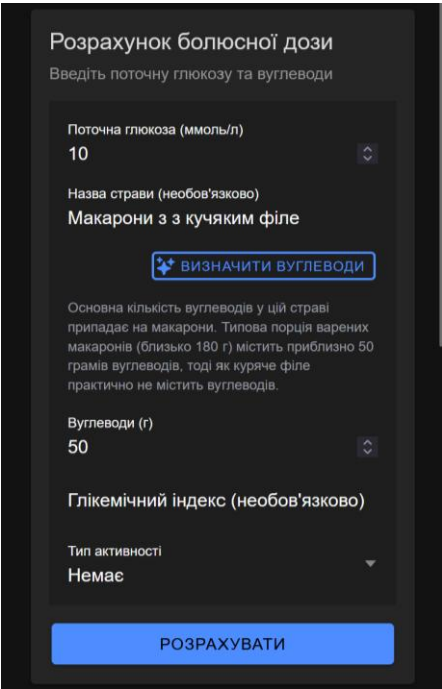


Рисунок 3.13 – Розрахунок вуглеводів за допомогою мовної моделі

Рисунок 3.14 – Вибір активності і інтенсивності на калькуляторі

Після натискання кнопки «Розрахувати» відображається картка з результатом: підсумкова доза, її складові, а саме вуглеводна частина, корекція та відрахування активного інсуліну IOB (рис. 3.15). Також виводиться поле для введення фактично введеної дози. Користувач може скоригувати дозу на власний розсуд перед збереженням.

Кнопка «Отримати пояснення від ШІ», що знаходиться нижче під розрахунками, викликає запит до мовної моделі. Мовна модель повертає пояснення розрахованої дози з урахуванням клінічного контексту; при виявленні ризику гіпоглікемії відображається термінове попередження без звернення до моделі.

Калькулятор болюсу

Рекомендована доза

3.2 Од.

Базова доза: 5.4 Од

Коефіцієнти: активність 60% × час 100% × тривалість 100%

На вуглеводи: 4.2 Од.

Корекція: +1.2 Од.

Активний інсулін (ІОВ): -0.0 Од.

Доза для введення (Од.)

3.2

Рекомендовано: 5.4 Од

ЗАПИСАТИ ДОЗУ

Рекомендація ШІ

Ваша розрахована доза інсуліну становить 5.4 одиниці. Вона включає 4.2 одиниці для засвоєння 50 грамів вуглеводів з вашої страви та 1.2 одиниці для корекції поточного підвищеного рівня глюкози 10.0 ммоль/л. Цей інсулін (прандіальний) вводиться перед їжею, щоб контролювати рівень цукру після неї. Після введення інсуліну та прийому їжі обов'язково контролюйте рівень глюкози, щоб переконатися, що він знижується до цільових показників і уникнути гіпоглікемії. Пам'ятайте, що ці рекомендації є індивідуальними і не замінюють консультації вашого лікаря.

Рисунок 3.15 – Розрахований болюс і рекомендація мовної моделі

Сторінка «Журнал їжі» містить всі попередні записи їжі, а сторінка «Журнал доз» містить форму для вводу ще однієї дози (наприклад болюсної або базальні ін'єкції) та список всіх попередніх ін'єкцій (рис. 3.16). Форма запитує кількість одиниць, які буде введено, тип ін'єкції (болюсна, базальна), рівень глюкози до введення та нотатки. Щоб створити запис треба натиснути кнопку «Записати».

Рисунок 3.16 – Сторінка журналу ін'єкцій

Наступна сторінка – сторінка налаштувань. Сторінка доступна в меню при натисканні на пункт «Профіль». На цій сторінці користувач може змінювати вигляд системи та редагувати персональні дані і клінічні параметри.

В блоці «Налаштування» користувач може змінювати мову та тему застосунку (рис. 3.17). При натисканні на поле «Мова» відкривається список з усіма можливими мовами. Наразі можливо встановити лише англійську або українську мови. При натисканні на поле «Тема» також відкривається список і надається можливість обрати світлу, темну і системну теми. За замовченням в застосунку встановлюється українська мова і системна тема.

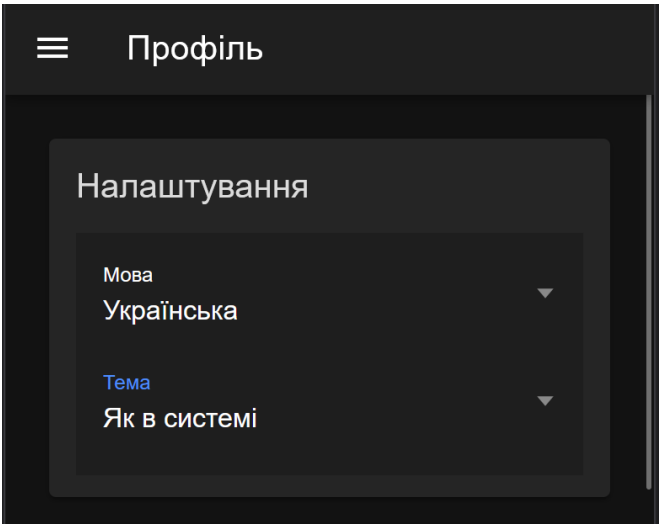


Рисунок 3.17 – Загальні налаштування застосунку

В блоці «Особиста інформація» користувач може змінити своє повне ім'я, але не має можливості змінювати електронну пошту, так як акаунт прив'язується до електронної пошти при реєстрації. Після блоку «Особиста інформація» знаходиться блок «Фізичні параметри», що відповідає параметрам ваги та зросту (рис. 3.18).

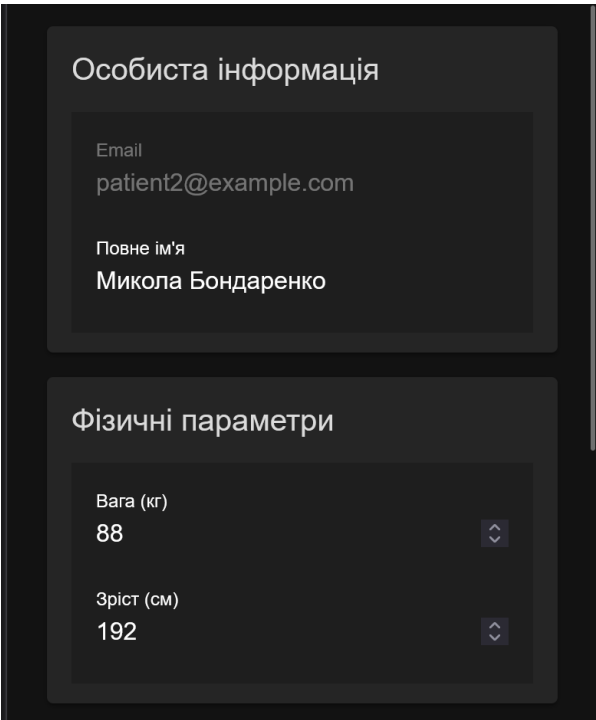


Рисунок 3.18 – Налаштування профілю та фізичних параметрів

В блоці «Параметри інсуліну» знаходять параметри, які мають основний вплив на роботу системи. В цьому блоці користувач може змінювати параметри коефіцієнт чутливості, інсуліно-вуглеводне співвідношення, тривалість дії інсуліну, коефіцієнт інсулін резистентності і типи болісного та базального інсулінів (рис. 3.19). Кожне поле має підказки під ним, що допомагає зрозуміти значення поля. Також поля мають спеціальні обмеження, що не дають ввести некоректні дані, наприклад текст або інші неправильні дані.

Профіль

Параметри інсуліну

Коефіцієнт чутливості (ммоль/л/Од)
3
 На скільки ммоль/л знижується глюкоза після 1 Од. Діапазон: 0.5–20

Інсуліново-вуглеводне співвідношення (г/Од)
12
 Скільки г вуглеводів покриває 1 Од. Діапазон: 1–50

Тривалість дії інсуліну (год)
4
 Зазвичай 3–6 год. Діапазон: 1–12

Коефіцієнт інсулінорезистентності
1
 1.0 = норма; >1.0 — знижена чутливість (більше інсуліну потрібно). Діапазон: 0.5–5.0

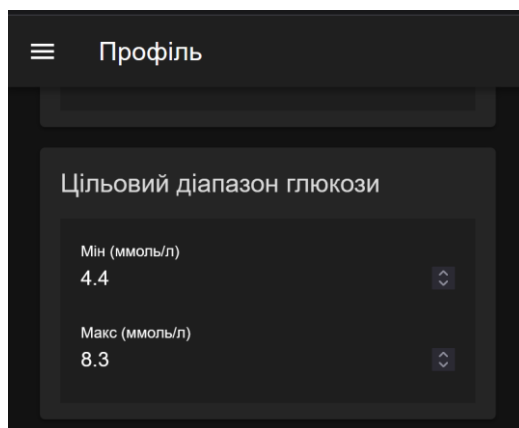
Час всмоктування вуглеводів (хв)
120
 Впливає на прогноз глюкози. Зазвичай 90–150 хв. Діапазон: 60–240

Базальний інсулін
Tresiba (Degludec)

Болісний інсулін
Apidra (Glulisine) — DIA 4 год

Рисунок 3.19 – Налаштування параметрів інсуліну

Наступний блок це «Цільовий діапазон глюкози», який визначає що для користувача є межі гіпо- і гіперглікемії (рис. 3.20). Поле «Мін» визначає нижню межу, нижче якої вважається гіпоглікемія для користувача. Аналогічно поле «Макс» визначає максимальну межу, вище якої вважається гіперглікемія.



Профіль

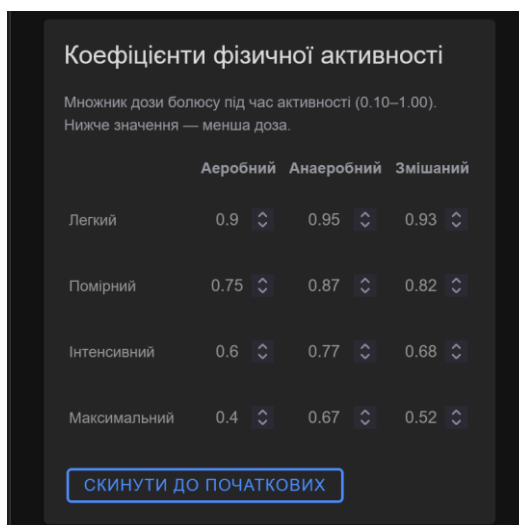
Цільовий діапазон глюкози

Мін (ммоль/л)
4.4

Макс (ммоль/л)
8.3

Рисунок 3.20 – Налаштування цільового діапазону інсуліну

Останній блок це блок «Коефіцієнти фізичної активності», які впливають на зниження дози при запланованій фізичній активності. Блок має вигляд таблиці, де значення відповідають комбінації параметрів на перехресті колонки та рядку (рис. 3.21). Нижче знаходиться кнопка скидання значень до тих що були до редагування.



Коефіцієнти фізичної активності

Множник дози болюсу під час активності (0.10–1.00).
Нижче значення — менша доза.

	Аеробний	Анаеробний	Змішаний
Легкий	0.9	0.95	0.93
Помірний	0.75	0.87	0.82
Інтенсивний	0.6	0.77	0.68
Максимальний	0.4	0.67	0.52

СКИНУТИ ДО ПОЧАТКОВИХ

Рисунок 3.21 – Налаштування коефіцієнтів фізичної активності

Для того щоб відбулось збереження змінених даних користувач повинен натиснути кнопку «Зберегти» внизу сторінки (рис. 3.22).

	Аеробний	Анаеробний	Змішаний
Легкий	0.9	0.95	0.93
Помірний	0.75	0.87	0.82
Інтенсивний	0.6	0.77	0.68
Максимальний	0.4	0.67	0.52

Рисунок 3.22 – Кнопка збереження налаштувань

При натисканні на кнопку «Зберегти» відбувається запит до серверу, який може займати деякий час через велику кількість перевірок правильності введених даних. При успішному збереженні буде показано повідомлення у вигляді тосту (рис. 3.23). В разі якщо на сервері буде знайдено помилкові або некоректні дані, то біля відповідного поля з'явиться повідомлення про помилку, а вигляд в застосунку буде зміщено до першого такого поля. Однак варіант, що застосунок поверне помилку малоймовірний, так як поля навмисно обмежують можливість введення некоректних даних.

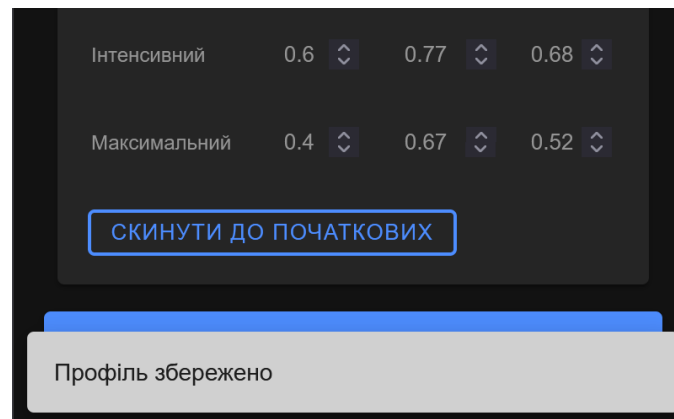


Рисунок 3.23 – Повідомлення про успішне збереження

Подібні повідомлення про збереження з’являються при натисканні на кожну з кнопок збереження даних в застосунку. Якщо виникають загальні помилки (наприклад сервер не відповідає) при обробці на сервері то над кнопкою буде виведено текст з помилкою.

Останній пункт в боковому меню це кнопка «Вихід» з червоним текстом. При натисканні на кнопку виходу ключі авторизації користувача будуть видалені і буде перенаправлено на сторінку автентифікації, що дозволить зробити повторний вхід в акаунт.

ВИСНОВКИ

У рамках кваліфікаційної роботи було розроблено інтелектуальний застосунок для розрахунку дози інсуліну для хворих на цукровий діабет I типу з функціями прогнозування глікемічних станів та формування персоналізованих рекомендацій.

Розроблений застосунок дозволяє:

- автоматизувати розрахунок болюсної дози інсуліну на основі рівня глюкози, кількості спожитих вуглеводів та індивідуальних параметрів пацієнта;
- прогнозувати динаміку рівня глюкози на горизонт до 180 хвилин з урахуванням активного інсуліну та вуглеводів;
- формувати персоналізовані клінічні рекомендації природною мовою на основі архітектури LLM з RAG;
- автоматично коригувати індивідуальні коефіцієнти ICR та ISF на основі накопичених даних пацієнта.

Для досягнення мети роботи застосовано такі методи та технології:

- математична модель розрахунку болюсної дози з урахуванням вуглеводного коефіцієнта, фактору чутливості, активного інсуліну та фізичної активності;
- архітектура RAG на основі моделі Gemini 3.5 Flash із семантичним пошуком за офіційними медичними протоколами через векторну базу pgvector;
- клієнт-серверна архітектура з використанням Spring Boot, Kotlin, Vue.js, Ionic та Docker.

На основі виконаної роботи зроблено такі висновки щодо розробленої системи:

- детермінований алгоритм розрахунку болюсної дози математично верифіковано на 50 клінічних сценаріях із нульовою середньою абсолютною похибкою;

- модуль оцінки вуглеводів на основі мовної моделі забезпечує точність 82% на вибірці страв типового українського раціону з МАЕ 5,2 г;
- модуль генерації рекомендацій пройшов клінічну перевірку на 50 сценаріях із дотриманням усіх критеріїв безпеки;
- критичні стани, зокрема гіпоглікемія, обробляються детермінованими засобами без залучення мовної моделі, що гарантує безпеку системи у граничних випадках;
- система повинна відповідати потребам пацієнтів із діабетом I типу, мінімізуючи ризик помилок розрахунку дози та когнітивне навантаження на користувача.

Розроблений застосунок є значущим результатом у сфері цифрової медицини та систем підтримки прийняття рішень, оскільки поєднує математичну точність детермінованих розрахунків з інтелектуальною гнучкістю великих мовних моделей для комплексного управління глікемією.

Результати роботи апробовано у вигляді тези доповіді під час Міжнародної науково-практичної конференції «Theoretical and Empirical Scientific Research: Concept and Trends» [22].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Improving Care and Promoting Health in Populations: Standards of Care in Diabetes – 2025. URL: <https://doi.org/10.2337/dc25-S001> (дата звернення 29.05.2026).
2. Insulin Analogs. URL: <https://diabetesteachingcenter.ucsf.edu/about-diabetes/type-2-diabetes/types-insulin-use-type-2-diabetes/insulin-analogs> (дата звернення 29.05.2026).
3. Hypoglycemia During Therapy of Diabetes. URL: <https://www.ncbi.nlm.nih.gov/books/NBK279100/> (дата звернення 30.05.2026).
4. Lyubchenko, V., & et al.. (2016). Digital image processing techniques for detection and diagnosis of fish diseases. International Journal of Advanced Research in Computer Science and Software Engineering, 6(7), 79-83.
5. Introduction to Glycemic Metrics. URL: [https://www.adces.org/education/danatech/glucose-monitoring/continuous-glucose-monitors-\(cgm\)/glycemic-metrics--an-overview](https://www.adces.org/education/danatech/glucose-monitoring/continuous-glucose-monitors-(cgm)/glycemic-metrics--an-overview) (дата звернення 30.05.2026).
6. Carbohydrate-to-Insulin Ratio in a Mediterranean Population of Type 1 Diabetic Patients on Continuous Subcutaneous Insulin Infusion Therapy. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC4604542/> (дата звернення 31.05.2026).
7. Zeleniy, O., Rudenko, D., Lyubchenko, V., & Lyashenko, V. (2022). Image Processing as an Analysis Tool in Medical Research. International Journal of Academic and Applied Research, 6(9), 1353141.
8. Boboyorov Sardor Uchqun o'g'li, Lyubchenko Valentin, & Lyashenko Vyacheslav. (2023). Image Processing Techniques as a Tool for the Analysis of Liver Diseases. Journal of Universal Science Research, 1(8), 223–233.
9. Lyashenko, V., Babker, A., Lyubchenko, V. (2017). Wavelet Analysis of Cytological Preparations Image in Different Color Systems. Open Access Library Journal, 4, 1-9.

10. Deep learning and regression approaches to forecasting blood glucose levels for type 1 diabetes. URL: <https://doi.org/10.1016/j.bspc.2021.102923> (дата звернення 31.05.2026).

11. Yakovleva, O., Matúšová, S., Liubchenko, V., & Maksimov, H. (2025). Innovative solutions based on AI in education, on the example of generating multimodal lecture notes. Scientific Journal of Bratislava University of Economics and Management «Public Administration and Regional Development, Economics, Management and Marketing», vol. 21, no. 1, pp.140–163.

12. Кульмінський, Я. К., & Любченко, В. А. (2025). Застосування мультимодальних моделей ШІ для автоматизованої оцінки інтерфейсів користувача. *Development of modern scientific technologies in the era of globalization: Proceedings of the 9th International scientific and practical conference* pp. 76–80.

13. Enhancing Large Language Model Reliability: Minimizing Hallucinations with Dual Retrieval-Augmented Generation Based on the Latest Diabetes Guidelines. URL: <https://doi.org/10.3390/jpm14121131> (дата звернення 31.05.2026).

14. Lyubchenko, V., Veretelnyk, K., Kots, P., & Lyashenko, V. (2024). Digital image segmentation procedure as an example of an NP-problem. Multidisciplinary Journal of Science and Technology, 4(4), 170-177.

15. Comparison of Three Interstitial Glucose Level Prediction Models for People with Diabetes. URL: <https://doi.org/10.3390/s23198269> (дата звернення 31.05.2026).

16. Calculating Insulin Dose. URL: <https://diabetesteachingcenter.ucsf.edu/about-diabetes/type-2-diabetes/use-insulin-type-2-diabetes/calculating-insulin-dose> (дата звернення 31.05.2026).

17. Mealtime Bolus Insulin Dose Timing in Children with Type 1 Diabetes: Reallife Data from a Tertiary Care Centre in Northern India. URL: <https://doi.org/10.4183/aeb.2021.528> (дата звернення 31.05.2026).

18. Boboyorov Sardor Uchqun o'g'li, Lyubchenko Valentin, & Lyashenko Vyacheslav. (2023). Pre-processing of digital images to improve the efficiency of liver fat analysis. *Multidisciplinary Journal of Science and Technology*, 3(1), 107–114.
19. Yevstratov, M., Lyubchenko, V., Amer, A. J., & Lyashenko, V. (2024). Color correction of the input image as an element of improving the quality of its visualization. *Technical science research in Uzbekistan*, 2(4), 79-88.
20. Ковтуненко, А. Р., Яковлева, О. В., Любченко, В. А., & Янголенко, О. В. (2020). Дослідження сумісного використання математичної морфології та згорткових нейронних мереж для вирішення задачі розпізнавання цінників.
21. Yakovleva, O., Kovtunenکو, A., Liubchenko, V., Honcharenko, V., & Kobylin, O. (2023). Face Detection for Video Surveillance-based Security System (COLINS-2023). In *CEUR Workshop Proceedings* (Vol. 3403). pp. 69-86.
22. Долгий, Є. О. (2026). Інтелектуальні застосунки для розрахунку дози інсуліну для хворих на цукровий діабет I типу. *Theoretical and empirical scientific research: Concept and trends: Proceedings of the X International Scientific and Practical Conference*. pp. 126-127.