

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Кафедра автоматизованих систем обробки інформації та управління

До захисту допущено:

В.о. завідувача кафедри

(підпис) Олександр ПАВЛОВ
(вл.ім'я, прізвище)

“ ____ ” _____ 2021 р.

Дипломний проєкт
на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інформаційні управляючі
системи та технології»
спеціальності 126 «Інформаційні системи та технології»**

на тему: *«Інформаційна система з контролю показників хворих на
цукровий діабет»*

Виконав:

студент IV курсу, групи ІС-73

Валь Олег Сергійович

(прізвище, ім'я, по батькові)

(підпис)

Керівник

доц., к.т.н., доц. Баклан Ігор Всеволодович

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

(підпис)

**Консультант з
графічної
документації**

доц., к.т.н., доц. Сперкач Майя Олегівна

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

(підпис)

Рецензент

доц., к.т.н., доц. Резніков Сергій Анатолійович

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

(підпис)

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент

(підпис)

Київ – 2021 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) інформатики та обчислювальної техніки
(повна назва)

Кафедра автоматизованих систем обробки інформації та управління

(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

Олександр ПАВЛОВ
(підпис) (вл.ім'я, прізвище)

“ ” 2021 р.

**ЗАВДАННЯ
на дипломний проєкт студенту**

Валь Олегу Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема проєкту «Інформаційна система з контролю показників хворих на цукровий діабет»,

керівник проєкту Баклан Ігор Всеволодович, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “ 11 ” 05 2021 р. № 1139-с

2. Термін подання студентом проєкту “04” червня 2021 року

3. Вихідні дані до проєкту

Технічне завдання

4. Зміст пояснювальної записки

1. Загальні положення: основні визначення та терміни, опис предметного середовища, огляд ринку програмних продуктів, постановка задачі

2. Інформаційне забезпечення: вхідні дані, вихідні дані, опис структури бази даних

3. Математичне забезпечення: змістовна та математична постановки задачі, обґрунтування та опис методу розв'язання

4. Програмне та технічне забезпечення: засоби розробки, вимоги до

технічного забезпечення, архітектура програмного забезпечення, побудова звітів

5. Технологічний розділ: керівництво користувача, методика випробувань програмного продукту

5. Перелік графічного матеріалу

1. Схема структурна варіантів використання

2. Схема бази даних

3. Схема структурна класів програмного забезпечення

4. Схема структурна компонентів програмного забезпечення

5. Схема структурна послідовності

6. Схема прикладу коду для розрахунку дозування інсуліну

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «7» квітня 2021 року

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Вивчення рекомендованої літератури	12.02.2021	
2.	Аналіз існуючих методів розв'язання задачі	20.02.2021	
3.	Постановка та формалізація задачі	03.03.2021	
4.	Розробка інформаційного забезпечення	16.03.2021	
5.	Алгоритмізація задачі	20.03.2021	
6.	Обґрунтування використовуваних технічних засобів	29.03.2021	
7.	Розробка програмного забезпечення	14.04.2021	
8.	Налагодження програми	22.04.2021	
9.	Виконання графічних документів	29.04.2021	
10.	Оформлення пояснювальної записки	04.05.2021	
11.	Подання ДП на попередній захист	14.05.2021	
12.	Подання ДП на основний захист	04.06.2021	
13.	Подання ДП рецензенту	07.06.2021	

Студент

Олег ВАЛЬ

Керівник

Ігор БАКЛАН

[illegible]

Пояснювальна записка до дипломного проєкту

на тему: «Інформаційна система з контролю показників хворих на цукровий
діабет»

Київ – 2021 року

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проєкту складається з п'яти розділів, містить 12 рисунків, 5 таблиць, 1 додаток, 15 джерел.

Дипломний проєкт присвячений розробці інформаційної системи з контролю показників хворих на цукровий діабет.

Основною ціллю розробки є допомога хворим на цукровий діабет слідкувати за станом свого здоров'я.

У розділі інформаційного забезпечення наведений опис структури вхідних та вихідних даних та структури бази даних.

У розділі математичного забезпечення наведений опис алгоритмів та математичної моделі, які використовуються в системі.

У розділі програмного забезпечення наведені діаграми класів, послідовності, компонентів, специфікацію функцій. Наведений опис архітектури системи, програмного та апаратного забезпечення.

У технологічному розділі описана інструкція користувача та наведено результати тестування інформаційної системи.

ДІАБЕТ, ІНСУЛІН, ГЛЮКОЗА, ТЕЛЕГРАМ, БОТ

ABSTRACT

Structure and scope of work. The explanatory note of the diploma project consists of five sections, containing 12 figures, 5 tables, 1 application, 15 sources.

The diploma project is devoted to the development of an information system for monitoring the performance of patients with diabetes.

The main goal of the development is to help diabetics monitor their health.

The information support section describes the structure of input and output data and the structure of the database.

The section of mathematical software describes the algorithms and mathematical models used in the system.

The software section provides diagrams of classes, sequences, components, specification of functions. The description of system architecture, software and hardware is given. The technological section describes the user manual and presents the test results of the information system.

DIABETES, INSULIN, GLUCOSE, TELEGRAM, BOT

					ДП 7301.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

ЗМІСТ

ВСТУП	6
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	7
1.1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА	7
1.1.1 Опис процесу діяльності	7
1.1.2 Опис функціональної моделі	7
1.2 ОГЛЯД ІСНУЮЧИХ АНАЛОГІВ.....	8
1.2.1 Діабет (Android, iOS, Apple Watch).....	9
1.2.2 Diabetes:M (Android, Android Wear, iOS, Apple Watch)	9
1.2.3 DiaMeter (Android, iOS)	9
1.2.4 ДіабетСтудія (Android, IOS)	10
1.3 ПОСТАНОВКА ЗАДАЧІ	11
1.3.1 Призначення розробки.....	11
1.3.2 Цілі та задачі розробки	11
Висновок до розділу	12
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ.....	13
2.1 ВХІДНІ ДАНІ	13
2.2 ВИХІДНІ ДАНІ.....	13
2.3 ОПИС СТРУКТУРИ БАЗИ ДАНИХ.....	14
Висновок до розділу	17
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ.....	18
3.1 ЗМІСТОВНА ПОСТАНОВКА ЗАДАЧІ.....	18
3.2 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ.....	18
3.3 ПСЕВДОКОД РОЗРАХУНКУ	20
3.4 ПРИКЛАД РОЗРАХУНКУ ДОЗИ ІНСУЛІНУ	21
Висновок до розділу	22
4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	23
4.1 ЗАСОБИ РОЗРОБКИ.....	23
4.2 ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ	23
4.2.1 Загальні вимоги	23
4.3 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	24

4.3.1	Діаграма класів.....	24
4.3.2	Діаграма послідовності.....	24
4.3.3	Діаграма компонентів.....	24
4.3.4	Специфікація функцій.....	24
4.4	ОПИС ЗВІТІВ.....	25
	Висновок до розділу	25
5	ТЕХНОЛОГІЧНИЙ РОЗДІЛ	26
5.1	КЕРІВНИЦТВО КОРИСТУВАЧА	26
5.2	ВИПРОБУВАННЯ ПРОГРАМНОГО ПРОДУКТУ	33
5.2.1	Мета випробувань.....	33
5.2.2	Загальні положення.....	33
5.2.3	Результати випробувань.....	33
	Висновок до розділу	34
	ЗАГАЛЬНІ ВИСНОВКИ	34
	ПЕРЕЛІК ПОСИЛАНЬ	36
	ДОДАТОК А.....	38

ВСТУП

Діабет — це хронічне захворювання, яке впливає на те, як тіло людини перетворює спожиту їжу на енергію.

Більша частина вжитої їжі, розщеплюється на глюкозу й викидається в кров. Коли рівень глюкози в крові підвищується, це сигналізує про те, що підшлункова залоза виділяє інсулін. Інсулін діє як ключ для проникнення крові з цукром у клітини організму для використання в якості енергії.

При діабеті організм або не виробляє достатньо інсуліну, або не може використовувати інсулін, який він виробляє так як слід. Коли інсуліну недостатньо або клітини перестають реагувати на інсулін, занадто багато цукру залишається в крові. З часом це може спричинити серйозні проблеми зі здоров'ям, такі як хвороби серця, втрата зору та хвороби нирок.

Ще немає ліків проти діабету, але дотримання дієти та активний спосіб життя дійсно можуть допомогти. Для того, щоб у хворих не було ускладнень, їм потрібно постійно стежити за станом свого здоров'я, регулярно вимірювати та фіксувати кількісні показники, що можуть використовуватися для діагностики медичними представниками.[1]

Практичне значення одержаних результатів. Розроблено інформаційну систему з контролю показників хворих на цукровий діабет.

Публікації. Результати роботи були опубліковані в 1 тезі доповіді на науково-технічній конференції [2].

					ДП 7301.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

Хворим на цукровий діабет потрібно постійно слідкувати за своїми показниками, щоб уникнути ускладнень свого захворювання та погіршення стану здоров'я в цілому.

У наш час дуже поширеним є використання месенджера "Telegram", в якому є можливість розробляти користувацьких ботів для повсякденного використання. Розроблена інформаційна система допоможе хворим вносити результати вимірів своїх показників та слідкувати за ними з будь-якого пристрою, який підтримує месенджер «Telegram»[3].

1.1.1 Опис процесу діяльності

У розробленій інформаційній системі буде відбуватися збір інформації про показники користувача й розрахунок потрібного дозування інсуліну, в залежності від даних про користувача.

1.1.2 Опис функціональної моделі

У розробленій інформаційній системі основним актором є користувач застосунку. Він має змогу вносити дані про рівень глюкози, введені ін'єкції та прийоми їжі, отримувати ці дані у вигляді графіку, а також розраховувати потрібну дозу інсуліну. Схему діаграми використання наведено в частині графічного матеріалу.

Відповідно до раніше зазначених варіантів використання було виявлено функціональні вимоги та встановлено їх пріоритетність, що відображено у таблиці 1.1.

					ДП 7301.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

Таблиця 1.1 – Функціональні вимоги

<i>Актор</i>	<i>Варіант використання</i>	<i>Функціональна вимога</i>	<i>Пріоритет</i>
Користувач застосунку	Запис показників до бази даних	Результати про виміри рівня глюкози, записи про введені ін'єкції інсуліну та записи про прийоми їжі повинні вноситись до бази даних і зберігатися в ній.	Важливий
	Заповнення профілю	Заповнення профілю користувача для розрахунку потрібної дози інсуліну.	Середній
	Розрахунок потрібної дози інсуліну	Використовуючи дані профілю розрахувати потрібну дозу інсуліну для ін'єкцій	Середній
	Отримання результатів вимірів	Користувач має змогу вивести дані про виміри рівню глюкози, введені ін'єкції та спожиті вуглеводи.	Важливий

1.2 Огляд існуючих аналогів

Оскільки проблема з цукровим діабетом у наш час є актуальною, то існує велика кількість аналогів із різноманітним функціоналом. Але навіть не дивлячись на це, не існує жодного чат-бота для Telegram, який допоможе хворим на цукровий діабет слідкувати за станом свого здоров'я. Нижче наведені аналоги серед мобільних додатків.

1.2.1 Діабет (Android, iOS, Apple Watch)

Один з найпопулярніших додатків у цій категорії, який допоможе вести щоденник контролю і розрахунку дози інсуліну, завдяки чому можна уникнути нападів гіпоглікемії.

Крім цього, є щоденник харчування й контроль рівня глюкози. Є можливість експорту даних для лікарів.

На iOS є інтеграція з додатком Здоров'я, куди додаються дані щодо кількості споживаних вуглеводів і рівень глюкози в крові. [4]

1.2.2 Diabetes:M (Android, Android Wear, iOS, Apple Watch)

У додатку є всі необхідні інструменти для діабетиків. Журнал контролю глюкози, інсуліну, харчування, приймання медикаментів, пам'ятки по ін'єкціях, замітки, категорії, нагадування та багато іншого.

На додаток в Diabetes: M є вбудована база їжі з можливістю додавання своїх продуктів. У ній можна відстежувати порції та кількість їжі. Є калькулятор вуглеводів і доз інсуліну. За допомогою докладних звітів можна відстежувати всі параметри, доступні в додатку. Є можливість імпорту звітів для лікаря.

У платній версії розблокується синхронізація між пристроями. [5]

1.2.3 DiaMeter (Android, iOS)

Додаток від російськомовного розробника. У DiaMeter можна легко відстежувати рівень цукру, кількість з'їдених хлібних одиниць, кількості ін'єкцій короткого і продовженого інсуліну. Додатково можна записувати дані про самопочуття та інше за допомогою нотаток.

На основі отриманих даних додаток відображає дані у вигляді зручних і зрозумілих графіків.

Є синхронізація даних між пристроями та корисні статті для діабетиків.

У платній версії з'являється експорт даних і аналізи цукру за тиждень і місяць. [6]

					ДП 7301.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2.4 ДіабетСтудія (Android, IOS)

Повністю безкоштовний додаток для діабетиків із першим типом діабету. ДіабетСтудія допоможе вести щоденник для відстеження основних показників: рівень інсуліну, цукор в крові, харчування, дієта, заняття спортом і багато іншого.

За допомогою графіків можна зрозуміти, як ваш раціон впливає на показники. Опція «Звіт для доктора» допоможе разом з лікарем детально стежити за ходом лікування.

У щоденнику є пошук записів за тегоми. Є можливість ведення записів батьками. Всі дані синхронізуються між пристроями. [7]

Дивлячись на вищезазначені аналоги, можемо побудувати таблицю переваг та недоліків: таблиця 1.2.

Таблиця 1.2 – Переваги та недоліки аналогів

<i>Функціонал / Аналог</i>	Діабет	Diabetes: M	DiaMeter	Діабет Студія	IC
<i>Контроль рівню глюкози</i>	Так	Так	Так	Так	Так
<i>Контроль введених ін'єкцій</i>	Так	Так	Так	Так	Так
<i>Контроль спожитих вуглеводів</i>	Так	Так	Так	Так	Так
<i>Розрахунок дози інсуліну</i>	Так	Так	Так	Так	Так

Продовження таблиці 1.2

<i>Функціонал / Аналог</i>	Діабет	Diabetes: M	DiaMeter	Діабет Студія	ІС
Кросплатформеність	Android, iOS, Apple Watch	Android, Android Wear, iOS, Apple Watch	Android, iOS	Android, IOS	Всі платформи, які підтримують Телеграм
Безкоштовність	Є платні функції	Є платні функції	Є платні функції	Є платні функції	Повністю безкоштовно
Підтримка української мови	Ні	Ні	Ні	Ні	Так

1.3 Постановка задачі

1.3.1 Призначення розробки

Призначення розробки — створення телеграм-бота, який надасть змогу хворим на цукровий діабет тримати свої показники, а саме рівень глюкози у нормі. Також забезпечити можливість стежити за своїм раціоном та введеними ін'єкціями інсуліну з будь-яких пристроїв, на яких встановлено телеграм.

1.3.2 Цілі та задачі розробки

Ціль розробки — вирішення проблеми контролю за показниками глюкози для хворих на цукровий діабет.

Задачі розробки:

- Спроекувати архітектуру серверної частини;
- Спроекувати базу даних;
- Розробити внесення даних профілю користувача, даних про вжиті вуглеводи, рівень глюкози та введені ін'єкції інсуліну;

- Реалізувати алгоритм обчислення потрібної дози інсуліну;
- Реалізувати можливість виводу даних у вигляді графіків.

Висновок до розділу

В даному розділі було розглянуто предметне середовище, сформульовані функціональні вимоги інформаційної системи та досліджені наявні аналоги. Також було сформульовано постановку задачі, призначення та цілі розробки.

					ДП 7301.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

Вхідні дані до інформаційної системи дані поступають через чат-бота в Telegram. Користувач заповнює свій профіль, результат вимірювання глюкози, запис про прийом їжі та запис про введену ін'єкцію інсуліну.

Профіль користувача містить наступні дані:

- Зріст;
- Вага;
- Вік;
- Стан хворого: Вперше виявлений цукровий діабет, більше року (добра компресія), більше року (нестійка компресія), декомпенсація, кетоацидоз, 3й триместр вагітності.

Результат вимірювання глюкози містить наступні дані:

- Кількість глюкози в крові;
- Коментар.

Запис про прийом їжі містить наступні дані:

- Кількість спожитих вуглеводів;
- Коментар.

Запис про введену ін'єкцію містить наступні дані:

- Введена доза інсуліну;
- Коментар.

2.2 Вихідні дані

На вихід подаються:

- Дані про користувача за запитом
- Результат розрахунку потрібної дози для ін'єкції інсуліну
- Дані записів про виміри рівня глюкози

- Дані записів про харчування
- Дані записів про введені ін'єкції інсуліну

2.3 Опис структури бази даних

У базі даних основними сутностями є користувач, дані вимірювань глюкози, дані про введені ін'єкції, дані про прийоми їжі. Отже, створені 3 таблиці.

В таблиці 2.1 показана структура БД у виді SQL:

Таблиця 2.1 – Структура БД та опис колонок

<i>Назва таблиці</i>	<i>Назва колонки</i>	<i>Тип даних</i>	<i>Опис</i>
User	id	serial	Первинний ключ, унікальний ідентифікатор
	telegramId	varchar(256) not null	Ідентифікатор користувача в телеграм
	height	double	Зріст користувача
	weight	double	Вага користувача

Продовження таблиці 2.1

<i>Назва таблиці</i>	<i>Назва колонки</i>	<i>Тип даних</i>	<i>Опис</i>
	age	int	Вік користува ча
	statusDise ase	varchar(256) not null	
EatingData	userId	integer references id column from User	Первинни й ключ, зовнішній ключ, ідентифіка тор користува ча
	carbohydr ates	integer	Кількість спожитих вуглеводів
	comment	varchar	Довільний коментар

Продовження таблиці 2.1

<i>Назва таблиці</i>	<i>Назва колонки</i>	<i>Тип даних</i>	<i>Опис</i>
	time	timestamp not null	Час прийому їжі
GlucoseData	userId	integer references id column from User	Первинни й ключ, зовнішній ключ, ідентифіка тор користува ча
	glucoseLe vel	double	Результат виміру рівня глюкози
	comment	varchar	Довільний коментар
	time	timestamp not null	Час виміру рівня глюкози

Продовження таблиці 2.1

<i>Назва таблиці</i>	<i>Назва колонки</i>	<i>Тип даних</i>	<i>Опис</i>
InjectionData	userId	integer references id column from User	Первинний ключ, зовнішній ключ, ідентифікатор користувача
	injectionDose	double	Введена доза інсуліну
	comment	varchar	Довільний коментар
	time	timestamp not null	Час ін'єкції інсуліну

Схема бази даних наведена в частині графічного матеріалу.

Висновок до розділу

У розділі інформаційного забезпечення було описано вхідні та вихідні дані інформаційної системи та спроектовано базу даних.

При проектуванні бази даних було створено 4 таблиці, які зберігають дані про користувачів та результати вимірювань.

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

Хворим на цукровий діабет першого типу потрібно регулярно вводити інсулін для підтримки нормального функціонування організму. Для того, щоб не нашкодити організму, потрібно правильно підібрати дозу вводимого інсуліну.

3.2 Математична постановка задачі

Нехай потрібно визначити добову дозу ін'єкції інсуліну для хворого на цукровий діабет.

Спочатку треба визначити коефіцієнт k (ОД/кг), який залежить від стану пацієнта s , залежність наведено в таблиці 3.1:

Таблиця 3.1 - Коефіцієнт в залежності від стану хворого

Стан	Коефіцієнт k (Од/кг)
Вперше виявлений ЦД типу 1	0.5
ЦД типу 1 тривалістю більше року в хорошій компенсації	0.6
ЦД типу 1 тривалістю більше року при нестійкій компенсації	0.7
ЦД типу 1 в ситуації декомпенсації	0.8
ЦД типу 1 в стані кетоацидозу	0.9
ЦД типу 1 в пубертатному періоді або в III триместрі вагітності	1

Також на коефіцієнт впливає надлишок, або нестача маси тіла. Для їх виявлення скористаємось індексом Кетле[8]. Для розрахунку індекса Кетле I знадобиться маса тіла m хворого та його зріст h .

$$I = \frac{m}{h^2}$$

Якщо значення індексу Кетле менше, ніж 18.5, то коефіцієнт k збільшується на 0.1. Якщо ж значення індексу Кетле більше, ніж 25, то коефіцієнт k зменшується на 0.1.

Щоб розрахувати добову потребу інсуліну D , потрібно помножити отриманий коефіцієнт на масу тіла хворого.

$$D = k * m$$

Добова потреба інсуліну подовженої дії рахується як половина від добової потреби з округленням до меншого.

$$IPD = \left[\frac{D}{2} \right]$$

Для розрахунку ранкової та вечірньої дози інсуліну подовженої дії треба взяти 60% та 40% відповідно.

Добова потреба інсуліну короткої дії рахується як добова доза інсуліну мінус добова доза інсуліну подовженої дії.

$$IKD = D - IPD$$

Інсулін короткої дії потрібно приймати перед їжею, враховуючи кількість вуглеводів та індивідуальні особливості організму, тому точне дозування однієї ін'єкції визначити не вийде.

3.3 Псевдокод розрахунку

```

float insulin_dose;
int age;
int height;
int weight;
float IMT=weight/height*height;
enum State{
FIRST_IDENTIFIED,
GOOD_COMPENSATION,
VOLATILE_COMPENSATION,
DECOMPENSATION,
KETOACIDOSIS,
PUBERTY}
State current = new State;
if(13<=age<=18){
    current=State.PUBERTY;
}
switch(current){
case FIRST_IDENTIFIED:
    insulin_dose=0.5;
    break;
case GOOD_COMPENSATION:
    insulin_dose=0.6;
    break;
case VOLATILE_COMPENSATION:
    insulin_dose=0.7;
    break;

```

```

case DECOMPENSATION:
    insulin_dose=0.8;
    break;
case KETOACIDOSIS:
    insulin_dose=0.9;
    break;
case PUBERTY:
    insulin_dose=1.0;
    break;
}
if(IMT<18.5){
    insulin_dose = insulin_dose + 0.1;
} else if(IMT>25){
    insulin_dose=insulin_dose-0.1;
}
insulin_dose=insulin_dose*weight;
int IPD = Math.floor(insulin_dose/2);
int IKD = insulin_dose-IPD;
    
```

3.4 Приклад розрахунку дози інсуліну

Хворий на цукровий діабет типу 1, хворий менше року. Вага 73 кг, зріст 176 см.

Добова потреба: $0,5 \text{ ОД} \times 73 \text{ кг} = 36,5 \text{ ОД}$ інсуліну.

Інсулін подовженої дії: 50% від 36.5 ОД = 18.25 (округляємо до 18 ОД):
перед сніданком – 10.8 ОД, на ніч 7.2 ОД.

Інсулін короткої дії: $36,5 - 18 = 18,5 \text{ ОД}$

Даний розрахунок - орієнтовний і потребує індивідуальної корекції, що проводиться під контролем рівня глікемії та споживання кількості вуглеводів у ХЕ.

Висновок до розділу

У розділі математичного забезпечення було розглянуто змістовну постановку задачі та математичні постановку задачі. Також було наведено псевдокод розрахунку потрібного дозування інсуліну та приклад розрахунку для заданого набору даних.

					ДП 7301.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

При розробці інформаційної системи було використано таке програмне забезпечення:

- Операційна система: Ubuntu 18.04 [9]
- Середовище розробки: Webstorm [10]
- Мови програмування: JavaScript [11]
- Платформа: Node.js [12]
- Фреймворки: Telegraf [13]
- ORM-бібліотека: Sequelize [14]
- База даних: Postgres [15]

4.2 Вимоги до технічного забезпечення

4.2.1 Загальні вимоги

Інформаційна система складається серверу, бази даних та клієнту Telegram.

Вимоги до серверу:

- ЦП з частотою 1 ГГц чи потужніший
- ОЗУ об'ємом 512 МБ чи більше
- 10 ГБ вільного місця на жорсткому диску

Можно налаштувати сервер, взявши таке ПЗ:

- Node.js 14
- Npm 6 [16]
- PostgreSQL з Postgis
- Ubuntu 16.04

Користувач повинен мати пристрій із встановленим додатком Telegram.

4.3 Архітектура програмного забезпечення

Архітектура програмного забезпечення складається з 3 компонент – серверу, бази даних та пристрою користувача.

4.3.1 Діаграма класів

Діаграма класів наведена в частині графічного матеріалу.

4.3.2 Діаграма послідовності

Схема структурна послідовності наведена у частині графічного матеріалу.

4.3.3 Діаграма компонентів

Схема структурна компонентів наведена у частині графічного матеріалу.

4.3.4 Специфікація функцій

У таблиці 4.1 описані специфікації функцій:

Таблиця 4.1 – Специфікації функцій

<i>Функція</i>	<i>Опис</i>
calculateCoefficient(status, age, height, weight): number	Обчислення коефіцієнту
calculateDailyDose(coefficient, weight): number	Визначення добової дози інсуліну
calculateIPD(dailyDose): {IPD, morningDose, eveningDose}	Визначення добової норми інсуліну подовженої дії, ранкової дози та вечірньої
calculateIKD(dailyDose, IPD): number	Визначення добової норми інсуліну короткої дії

4.4 Опис звітів

Результатом роботи є розрахована норма інсуліну та графіки, які відображають дані, записані користувачем.

Висновок до розділу

У даному розділі описано програмне забезпечення інформаційної системи. Платформа, на якій написано інформаційну систему – Node.js, СУБД – Postgres. Було описано архітектуру інформаційної системи, наведено діаграму класів та специфікації функцій класів.

					ДП 7301.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Керівництво користувача

Для того, щоб почати користуватися ынформаційною системою, треба зайти в телеграм та за тегом @diplom_diabet_bot перейти до телеграм-боту. При початку роботи з ботом потрібно буде заповнити профіль користувача. На рисунку 5.1 наведено приклад заповнення.

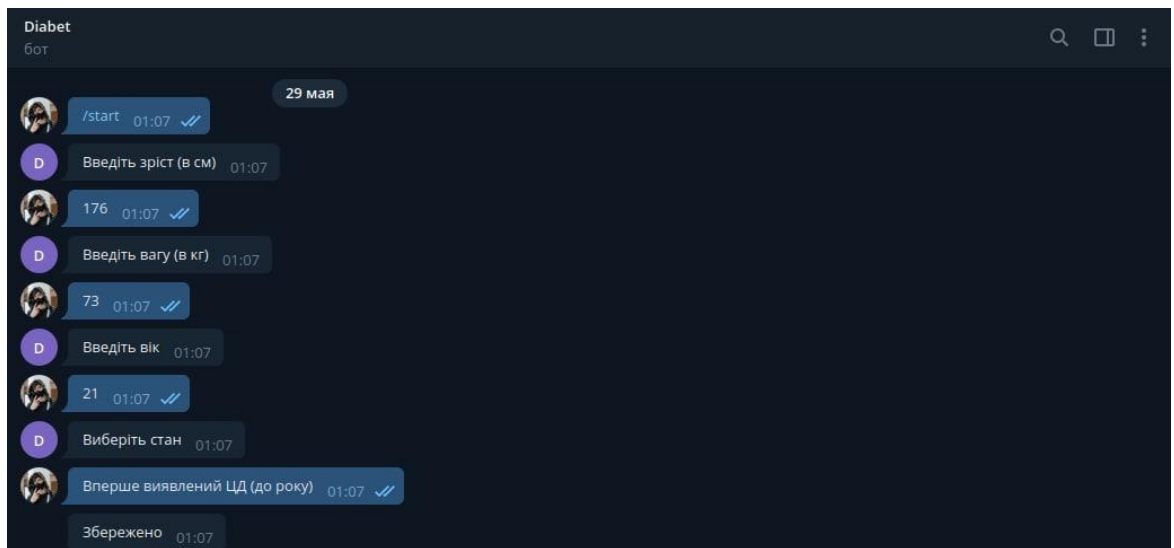


Рисунок 5.1 – Заповнення профілю

Після заповнення профілю з'явиться головне меню, яке показано на рисунку 5.2.

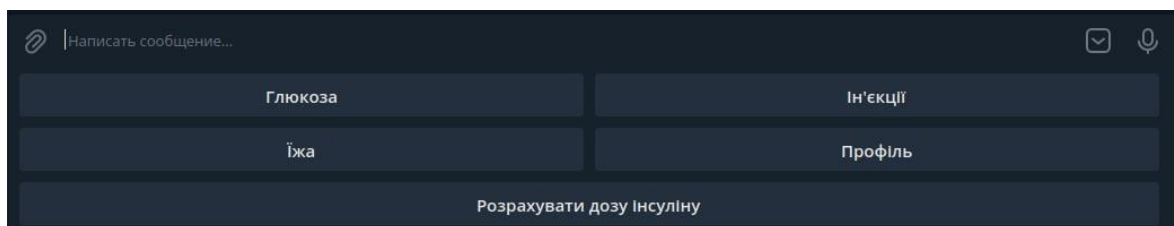


Рисунок 5.2 – Головне меню

При натисканні на кнопку «Глюкоза» в головному меню відбудеться перехід до меню, яке надасть можливість внести дані про вимірювання рівня глюкози або вивести графік вимірювань. Дане меню наведено на рисунку 5.3.

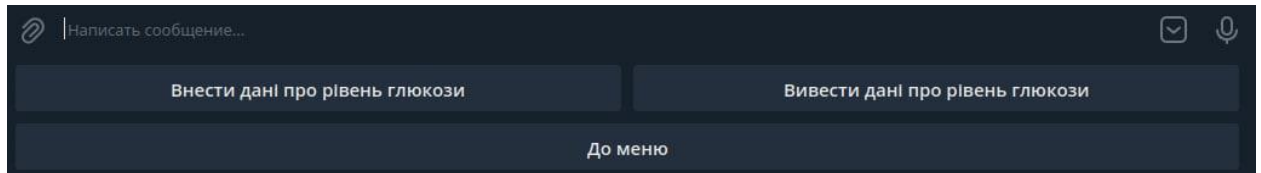


Рисунок 5.3 – Меню глюкози

Приклад внесення даних про вимірювання рівня глюкози у крові наведено на рисунку 5.4.

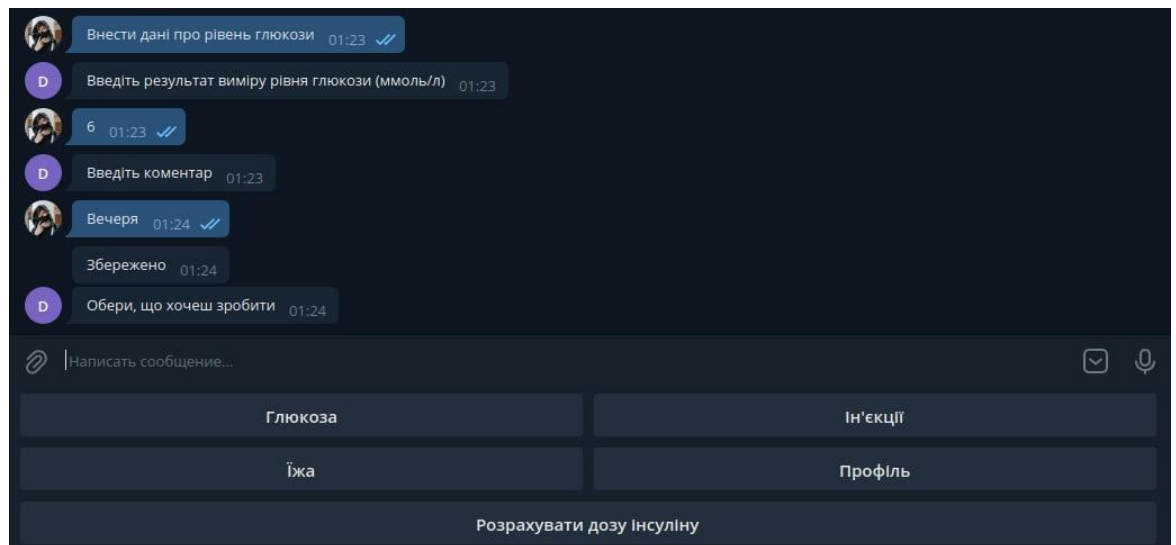


Рисунок 5.4 – Внесення результатів вимірювання глюкози

Приклад виведення даних про вимірювання глюкози у крові наведено на рисунку 5.5.

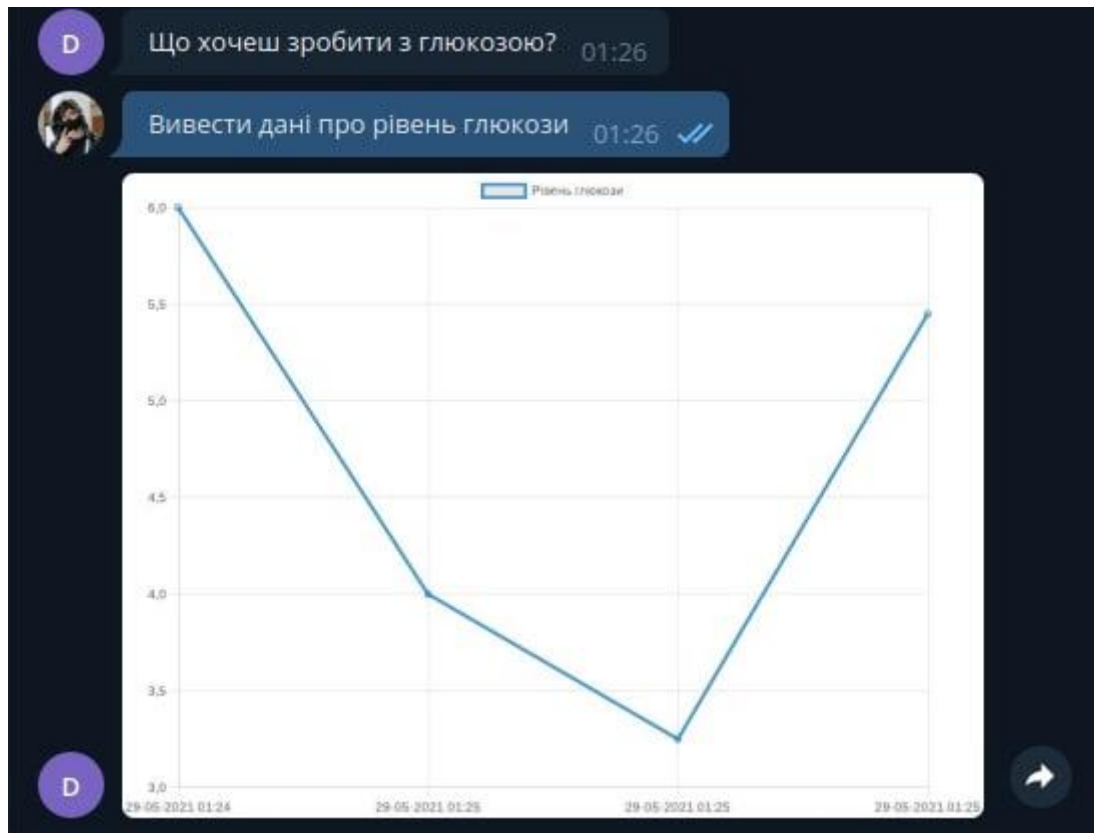


Рисунок 5.5 – Виведення даних про вимірювання рівня глюкози

При натисканні на кнопку «Ін'єкції» в головному меню відбудеться перехід до меню, яке надасть можливість внести дані про дозування введених ін'єкцій або вивести графік дозувань. Дане меню наведено на рисунку 5.6.

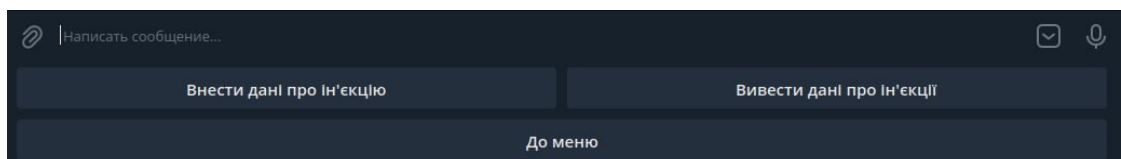


Рисунок 5.6 – Меню ін'єкцій

Приклад внесення даних про введені ін'єкції наведено на рисунку 5.7.

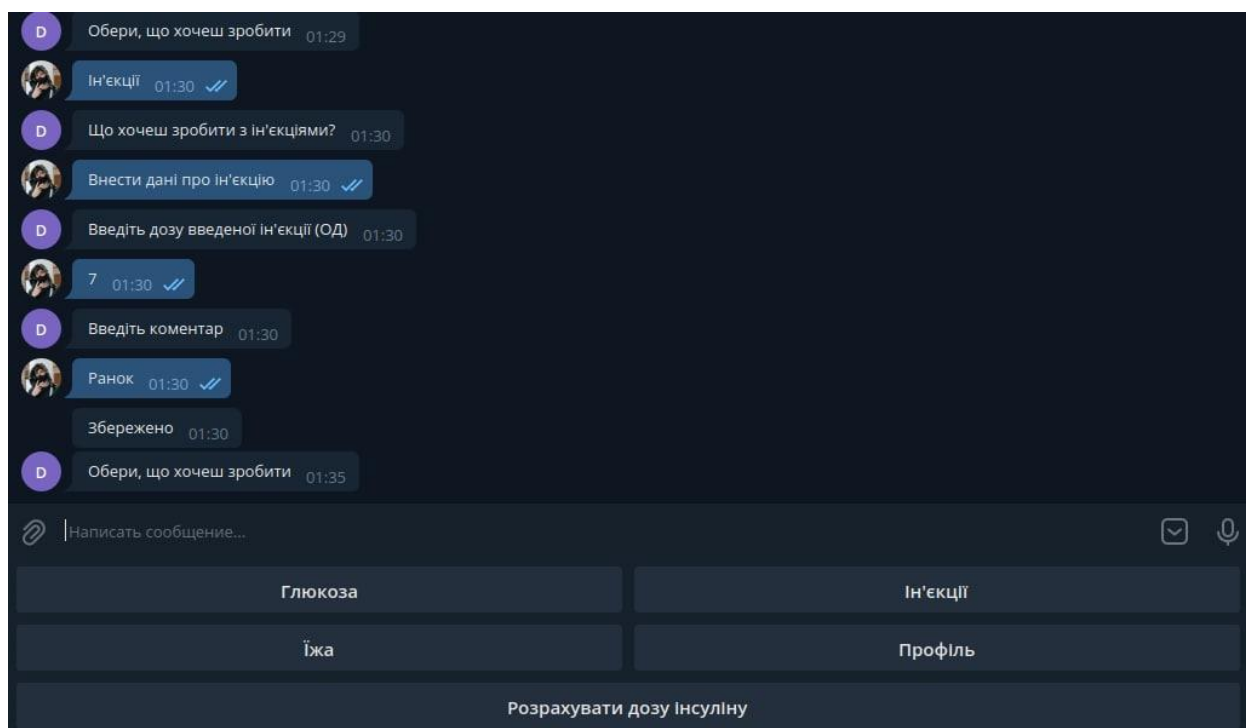


Рисунок 5.7 – Внесення результатів про введену ін'єкцію

Приклад виведення даних про введені ін'єкції наведено на рисунку 5.8.

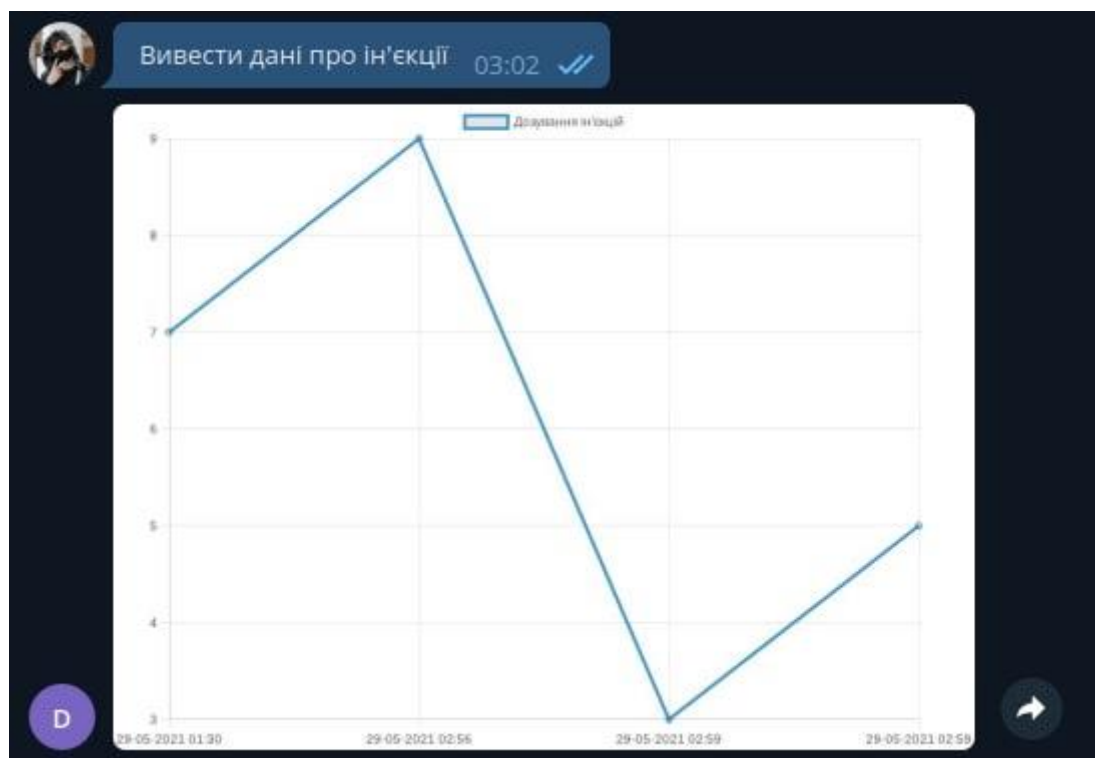


Рисунок 5.8 – Виведення даних про введені ін'єкції

При натисканні на кнопку «Їжа» в головному меню відбудеться перехід до меню, яке надасть можливість внести дані про спожиті вуглеводи за прийом їжі або вивести графік спожитих вуглеводів. Дане меню наведено на рисунку 5.9.

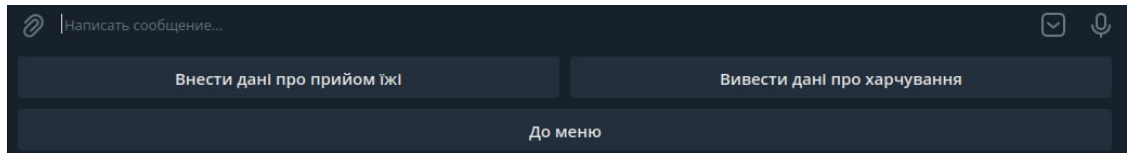


Рисунок 5.9 – Меню їжі

Приклад внесення даних про спожиті вуглеводи наведено на рисунку 5.10.

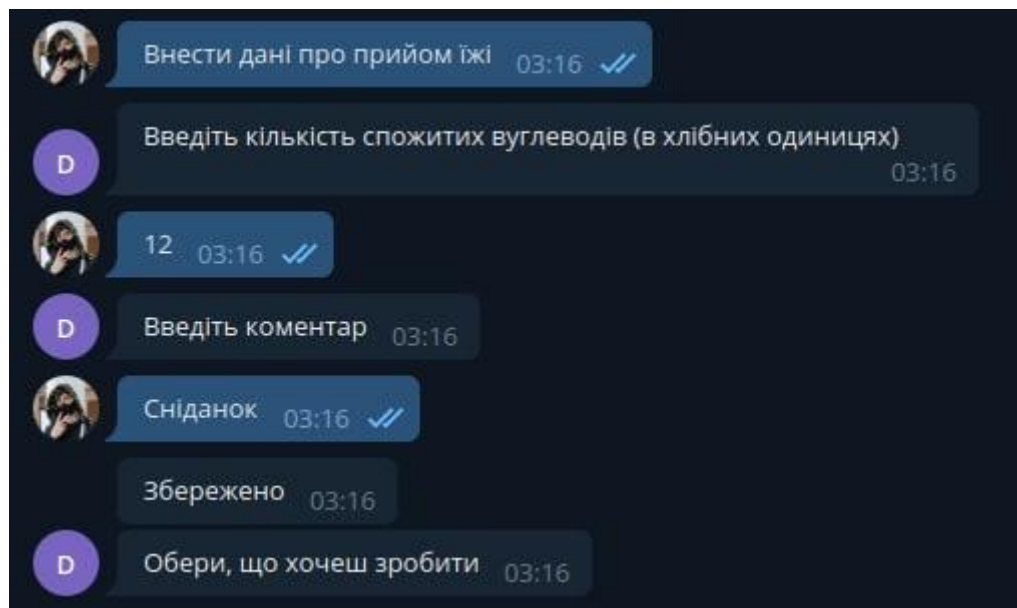


Рисунок 5.10 – Внесення результатів про спожиті вуглеводи

Приклад виведення даних про спожиті вуглеводи наведено на рисунку 5.11.

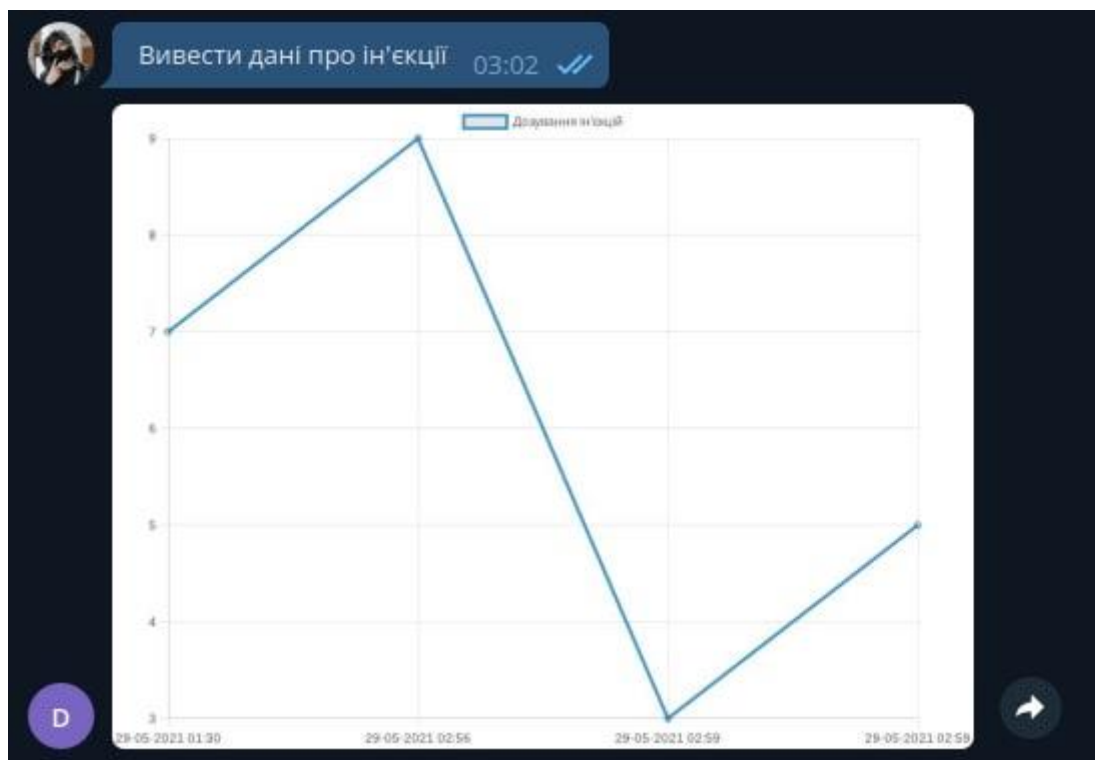


Рисунок 5.11 – Виведення даних про спожиті вуглеводи

При натисканні на кнопку «Розрахувати дозу інсуліну» відбудеться розрахунок добової норми інсуліну, долі інсуліну довгої та короткої дії, та ранкова та вечірня доза інсуліну довгої дії. Розрахунок відбувається з використанням даних профілю користувача. Результат розрахунку наведено на рисунку 5.12.

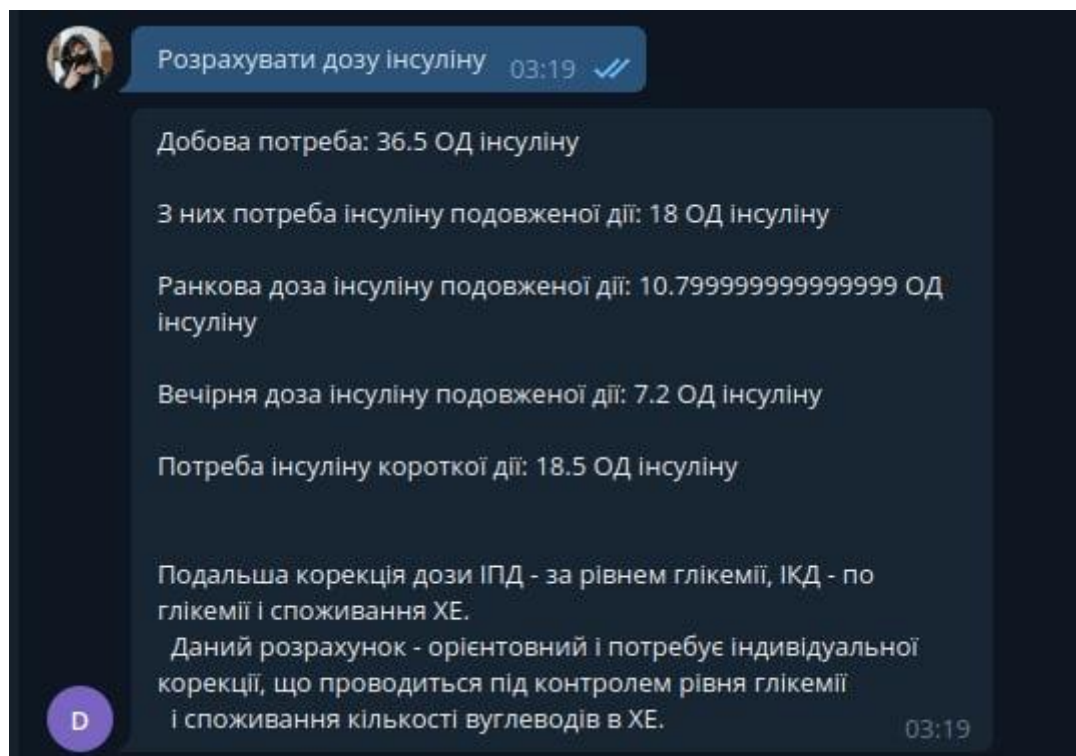


Рисунок 5.12 – Результат розрахунку дози інсуліну

5.2 Випробування програмного продукту

5.2.1 Мета випробувань

Метою випробувань є перевірка відповідності функцій розробленого програмного забезпечення для інформаційної системи з контролю показників хворих на цукровий діабет вимогам технічного завдання.

5.2.2 Загальні положення

Основою для випробувань являються наступні документи:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

5.2.3 Результати випробувань

Метою тестування була перевірка працездатності основних функцій інформаційної системи. Було перевірено такі функції:

- Внесення та вивід даних про рівень глюкози в крові
- Внесення та вивід даних про введені дози інсуліну
- Внесення та вивід даних про прийоми їжі
- Внесення та вивід даних профілю
- Розрахунок потрібного дозування інсуліну, використовуючи дані профілю користувача

Висновок до розділу

У даному розділі було надано детальну інструкцію з користування розробленою інформаційною системою. За допомогою скріншотів з поясненнями було продемонстровано весь функціонал розробленого програмного забезпечення.

Також було наведено мету випробовувань та загальні положення.

Додаток було протестовано. Тестування пройшло успішно, тому розроблене програмне забезпечення відповідає вимогам та в змозі розв'язувати поставлені задачі.

ЗАГАЛЬНІ ВИСНОВКИ

Результатом виконання дипломного проєкту є інформаційна система з контролю показників хворих на цукровий діабет.

Було проаналізовано актуальність розробленого програмного продукту, визначено існуючі підходи до розробки, а також цілі та призначення розробки, описано функціональну модель системи, предметне середовище та процес діяльності.

Було наведено вхідні та вихідні дані та структуру сутностей бази даних.

Було сформульовано змістовну та математичну постановку задачі, яка полягає в розрахунку потреби вводимого інсуліну користувачеві системи, виходячи з даних профілю.

В дипломному проєкті також було описано технології, які використовувались для розробки програмного продукту. Було обрано реляційну базу даних Postgres в якості бази даних. Було використано мову програмування JavaScript на платформі Node.js. Було розглянуто описи класів, які входять до архітектури розробки та наведено вимоги до технічного забезпечення.

					ДП 7301.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

В дипломному проєкті також присутнє керівництво користувача, в якому описано, як користувачеві взаємодіяти з програмним продуктом. Було протестовано основні функції програмного продукту.

Розроблений програмний продукт зможе спростити життя багатьох людей, які хворіють на цукровий діабет.

					ДП 7301.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

ПЕРЕЛІК ПОСИЛАНЬ

1. Глобальный доклад по диабету [Global report on diabetes]. Женева: Всемирная организация здравоохранения; 2018. Лицензия: CC BY-NC-SA 3.0 IGO. Данные каталогизации перед публикацией (CIP).
Данные CIP доступны по ссылке:
<http://apps.who.int/iris/>
2. Валь О.С., Контроль показників хворих на цукровий діабет // Матеріали VI всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2021) – м. Київ.: НТУУ «КПІ ім. Ігоря Сікорського», 22-23 квітня 2021 р.
3. Telegram [Електронний ресурс] – Режим доступу до ресурсу:
<https://telegram.org/>
4. Діабет [Електронний ресурс] - Режим доступу до ресурсу:
<https://play.google.com/store/apps/details?id=ru.hintsolutions.diabets>
5. Diabetes-m [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.diabetes-m.com/>
6. DiaMeter [Електронний ресурс] - Режим доступу до ресурсу:
<https://diameterapp.ru/>
7. ДіабетСтудія [Електронний ресурс] - Режим доступу до ресурсу:
<https://diabetes-studio.com>
8. Индекс Кетле [Електронний ресурс] - Режим доступу до ресурсу:
<https://gnicpm.ru/articles/zdorovyj-obraz-zhizni/pokazateli-massy-tela.html>
9. Ubuntu [Електронний ресурс] - Режим доступу до ресурсу:
<https://ubuntu.com/>
10. Webstorm [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.jetbrains.com/ru-ru/webstorm/>
11. JavaScript [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.javascript.com/>

12.Node.js [Електронний ресурс] - Режим доступу до ресурсу:

<https://nodejs.org>

13.Telegraf [Електронний ресурс] - Режим доступу до ресурсу:

<https://telegraf.js.org/>

14.Sequelize [Електронний ресурс] - Режим доступу до ресурсу:

<https://sequelize.org/>

15.Postgres [Електронний ресурс] - Режим доступу до ресурсу:

<https://www.postgresql.org/>

16.npm [Електронний ресурс] - Режим доступу до ресурсу:

<https://www.npmjs.com/>

Додаток А

Тексти програмного коду

**«Інформаційна система з контролю показників хворих на цукровий
діабет»**

(Найменування програми (документа))

DVD-R

(Вид носія даних)

44 арк, 14.3 Mb

(Обсяг програми (документа) , арк., Кб)

Київ – 2021 року

					ДП 7301.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

src/index.js

```
import telegram from './telegram';
```

```
import models from './data/models';
```

```
process
```

```
.on('unhandledRejection', (reason, p) => {  
  console.error('Unhandled Rejection at:', p, 'reason:', reason);  
})
```

```
.on('uncaughtException', (err) => {  
  console.error(err, 'Uncaught Exception thrown', err.stack);  
  process.exit(1);  
});
```

```
(async () => {  
  await models.sync();  
  await telegram();  
  console.info('Bot started successfully');  
})();
```

					ДП 7301.00.000 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

src/config

```
import dotenv from 'dotenv';
```

```
dotenv.config();
```

```
// eslint-disable-next-line
```

```
globalThis.__DEV__ = process.env.NODE_ENV !== 'production';
```

```
export const token = process.env.TELEGRAM_BOT_TOKEN;
```

```
export const databaseUrl = process.env.DATABASE_URL;
```

					ДП 7301.00.000 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

src/telegram.js

```
import { Telegraf } from 'telegraf';
import setupCommands from './data/commands';
import { token } from './config';

let bot = null;

export default async () => {
  if (!token) {
    throw new Error('No telegram bot key supplied');
  }
  try {
    bot = new Telegraf(token);

    setupCommands(bot);

    await bot.launch();
  } catch (e) {
    console.error(e);
  }
};
```

					ДП 7301.00.000 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

src/data/sequelize.js

```
import Sequelize from 'sequelize';
import { databaseUrl } from '../config';

const sequelize = new Sequelize(databaseUrl, {
  logging: false,
  dialect: 'postgres',
  define: {
    freezeTableName: true,
  },
});

export default sequelize;
```

					ДП 7301.00.000 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

src/data/database.js

```
require('dotenv').config();
```

```
module.exports = {  
  development: {  
    use_env_variable: 'DATABASE_URL',  
    dialect: 'postgres',  
  },  
  production: {  
    use_env_variable: 'DATABASE_URL',  
    dialect: 'postgres',  
  },  
};
```

					ДП 7301.00.000 ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

src/data/constants.js

```

export const glucose = 'Глюкоза';
export const injection = 'Ін'єкції';
export const food = 'Їжа';
export const profile = 'Профіль';
export const toMenu = 'До меню';

export const insulinDose = 'Розрахувати дозу інсуліну';

export const profileStartIn = 'Заповнити профіль';
export const profileStartOut = 'Вивести дані профілю';

export const glucoseStartIn = 'Внести дані про рівень глюкози';
export const glucoseStartOut = 'Вивести дані про рівень глюкози';

export const injectionStartIn = 'Внести дані про ін'єкцію';
export const injectionStartOut = 'Вивести дані про ін'єкції';

export const foodStartIn = 'Внести дані про прийом їжі';
export const foodStartOut = 'Вивести дані про харчування';

export const glucoseWizardId = 'glucose-wizard';
export const foodWizardId = 'food-wizard';
export const injectionWizardId = 'injection-wizard';
export const profileWizardId = 'profile-wizard';

export const statusEnum = {
  firstIdentified: 'Вперше виявлений ЦД (до року)',
  goodCompensation: 'Хороша компенсація',
  volatileCompensation: 'Нестійка компенсація',
  decompensation: 'Декомпенсація',
  ketoacidosis: 'Кетоацидоз',
  pregnancy: '3 триместр вагітності',
};

```

src/data/models/User.js

```
import Sequelize from 'sequelize';
import Model from '../sequelize';
```

```
const User = Model.define(
  'User',
  {
    id: {
      type: Sequelize.INTEGER,
      primaryKey: true,
      autoIncrement: true,
    },
    telegramId: {
      type: Sequelize.INTEGER,
      allowNull: false,
      unique: true,
    },
    height: {
      type: Sequelize.DOUBLE(4, 2),
    },
    weight: {
      type: Sequelize.DOUBLE(4, 2),
    },
    age: {
      type: Sequelize.INTEGER,
    },
    statusDisease: {
      type: Sequelize.STRING(1024),
    },
  },
  { timestamps: false },
);
```

```
User.assert = async (telegramId) => {
  const [user] = await User.upsert({ telegramId }, { returning: true });
  return user;
};
```

```
User.updateUser = async (userId, height, weight, age, statusDisease) => {
  try {
    await User.update(
      { height, weight, age, statusDisease }, // what going to be updated
      { where: { id: userId } }, // where clause
    );
  } catch (error) {
    console.error(error);
  }
};
```

```
);  
} catch (error) {  
    // error handling  
}  
};
```

export default User;

src/data/models/User.js

```
import Sequelize from 'sequelize';
import Model from '../sequelize';
```

```
const User = Model.define(
  'User',
  {
    id: {
      type: Sequelize.INTEGER,
      primaryKey: true,
      autoIncrement: true,
    },
    telegramId: {
      type: Sequelize.INTEGER,
      allowNull: false,
      unique: true,
    },
    height: {
      type: Sequelize.DOUBLE(4, 2),
    },
    weight: {
      type: Sequelize.DOUBLE(4, 2),
    },
    age: {
      type: Sequelize.INTEGER,
    },
    statusDisease: {
      type: Sequelize.STRING(1024),
    },
  },
  { timestamps: false },
);
```

```
User.assert = async (telegramId) => {
  const [user] = await User.upsert({ telegramId }, { returning: true });
  return user;
};
```

```
User.updateUser = async (userId, height, weight, age, statusDisease) => {
  try {
    await User.update(
      { height, weight, age, statusDisease }, // what going to be updated
      { where: { id: userId } }, // where clause
    );
  } catch (error) {
    console.error(error);
  }
};
```

```
);  
} catch (error) {  
    // error handling  
}  
};
```

```
export default User;
```

src/data/models/InjectionData.js

```
import Sequelize from 'sequelize';
import Model from '../sequelize';
```

```
const InjectionData = Model.define(
  'InjectionData',
  {
    id: {
      type: Sequelize.INTEGER,
      primaryKey: true,
      autoIncrement: true,
    },
    injectionDose: {
      type: Sequelize.DOUBLE(4, 2),
      allowNull: false,
    },
    comment: {
      type: Sequelize.STRING(1024),
    },
  },
  { updatedAt: false },
);
```

```
InjectionData.add = async (userId, injectionDose, comment) => {
  await InjectionData.upsert({ userId, injectionDose, comment });
};
```

```
export default InjectionData;
```

src/data/models/GlucoseData.js

```
import Sequelize from 'sequelize';
import Model from '../sequelize';

const GlucoseData = Model.define(
  'GlucoseData',
  {
    id: {
      type: Sequelize.INTEGER,
      primaryKey: true,
      autoIncrement: true,
    },
    glucoseLevel: {
      type: Sequelize.DOUBLE(4, 2),
      allowNull: false,
    },
    comment: {
      type: Sequelize.STRING(1024),
    },
  },
  { updatedAt: false },
);

GlucoseData.add = async (userId, glucoseLevel, comment) => {
  await GlucoseData.upsert({ userId, glucoseLevel, comment });
};

export default GlucoseData;
```

src/data/models/EatingData.js

```
import Sequelize from 'sequelize';
import Model from '../sequelize';
```

```
const EatingData = Model.define(
  'EatingData',
  {
    id: {
      type: Sequelize.INTEGER,
      primaryKey: true,
      autoIncrement: true,
    },
    carbohydrates: {
      type: Sequelize.INTEGER,
      allowNull: false,
    },
    comment: {
      type: Sequelize.STRING(1024),
    },
  },
  { updatedAt: false },
);
```

```
EatingData.add = async (userId, carbohydrates, comment) => {
  await EatingData.upsert({ userId, carbohydrates, comment });
};
```

```
export default EatingData;
```

src/data/models/index.js

```
import sequelize from '../sequelize';
import User from './User';
import EatingData from './EatingData';
import GlucoseData from './GlucoseData';
import InjectionData from './InjectionData';

User.hasMany(EatingData, {
  foreignKey: 'userId',
  as: 'eatings',
});

User.hasMany(GlucoseData, {
  foreignKey: 'userId',
  as: 'glucoces',
});

User.hasMany(InjectionData, {
  foreignKey: 'userId',
  as: 'injections',
});

const sync = (...args) => sequelize.sync(...args);

export default { sync };

export { EatingData, User, GlucoseData, InjectionData };
```

src/data/plot/createFoodPlot.js

```

import moment from 'moment';
import { EatingData } from '../models';

const { ChartJSNodeCanvas } = require('chartjs-node-canvas');

const chartJSNodeCanvas = new ChartJSNodeCanvas({
  type: 'pdf',
  width: 800,
  height: 600,
});

const getEatingData = (userId) =>
  EatingData.findAll({ where: { userId }, order: ['createdAt'] });

export const createFoodPlot = async (ctx) => {
  const food = await getEatingData(ctx.session.user.id);
  console.table(
    food.map((f) => ({
      x: moment(f.createdAt).format('DD-MM-YYYY HH:mm'),
      y: f.carbohydrates,
    })),
  );
  console.log('here');

  const image = chartJSNodeCanvas.renderToBufferSync({
    type: 'line',
    data: {
      labels: food.map((f) => moment(f.createdAt).format('DD-MM-YYYY HH:mm')),
      datasets: [
        {
          data: food.map((g) => ({
            x: moment(g.createdAt).format('DD-MM-YYYY HH:mm'),
            y: g.carbohydrates,
          })),
          label: 'Рівень вуглеводів',
          borderColor: '#3e95cd',
          fill: false,
        },
      ],
    },
  });
};

```

```
console.log('hi!');  
  
return ctx.replyWithPhoto({ source: image });  
};
```

src/data/plot/createInjectionPlot.js

```

import moment from 'moment';
import { InjectionData } from '../models';

const { ChartJSNodeCanvas } = require('chartjs-node-canvas');

const chartJSNodeCanvas = new ChartJSNodeCanvas({
  type: 'pdf',
  width: 800,
  height: 600,
});

const getInjectionData = (userId) =>
  InjectionData.findAll({ where: { userId }, order: ['createdAt'] });

export const createInjectionPlot = async (ctx) => {
  const gl = await getInjectionData(ctx.session.user.id);
  console.table(
    gl.map((g) => ({
      x: moment(g.createdAt).format('DD-MM-YYYY HH:mm'),
      y: g.injectionDose,
    })),
  );
  console.log('here');

  const image = chartJSNodeCanvas.renderToBufferSync({
    type: 'line',
    data: {
      labels: gl.map((g) => moment(g.createdAt).format('DD-MM-YYYY HH:mm')),
      datasets: [
        {
          data: gl.map((g) => ({
            x: moment(g.createdAt).format('DD-MM-YYYY HH:mm'),
            y: g.injectionDose,
          })),
          label: "Дозування ін'єкцій",
          borderColor: '#3e95cd',
          fill: false,
        },
      ],
    },
  );
};

```

```
console.log('hi!');  
  
return ctx.replyWithPhoto({ source: image });  
};
```

src/data/plot/createGlucosePlot.js

```

import moment from 'moment';
import { GlucoseData } from '../models';

const { ChartJSNodeCanvas } = require('chartjs-node-canvas');

const chartJSNodeCanvas = new ChartJSNodeCanvas({
  type: 'pdf',
  width: 800,
  height: 600,
});

const getGlucoseData = (userId) =>
  GlucoseData.findAll({ where: { userId }, order: ['createdAt'] });

export const createGlucosePlot = async (ctx) => {
  const gl = await getGlucoseData(ctx.session.user.id);
  console.table(
    gl.map((g) => ({
      x: moment(g.createdAt).format('DD-MM-YYYY HH:mm'),
      y: g.glucoseLevel,
    })),
  );
  console.log('here');

  const image = chartJSNodeCanvas.renderToBufferSync({
    type: 'line',
    data: {
      labels: gl.map((g) => moment(g.createdAt).format('DD-MM-YYYY HH:mm')),
      datasets: [
        {
          data: gl.map((g) => ({
            x: moment(g.createdAt).format('DD-MM-YYYY HH:mm'),
            y: g.glucoseLevel,
          })),
          label: 'Рівень глюкози',
          borderColor: '#3e95cd',
          fill: false,
        },
      ],
    },
  );
};

```

```
console.log('hi!');  
  
return ctx.replyWithPhoto({ source: image });  
};
```

src/data/commands/index.js

```
import { Scenes, session } from 'telegraf';
import menuHandler from './keyboards/menu';
import {
  glucose,
  glucoseStartOut,
  glucoseWizardId,
  glucoseStartIn,
  toMenu,
  food,
  foodStartIn,
  foodStartOut,
  foodWizardId,
  injection,
  injectionStartIn,
  injectionStartOut,
  injectionWizardId,
  profile,
  profileStartIn,
  profileWizardId,
  profileStartOut,
  insulinDose,
} from './constants';
import { glucoseStart } from './keyboards/glucose';
import cancel from './cancel';
import upsertUser from './upsertUser';
import glucoseWizard from './wizards/glucoseWizard';
import foodWizard from './wizards/foodWizard';
import injectionWizard from './wizards/injectionWizard';
```

					ДП 7301.00.000 ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

```

import { createGlucosePLot } from '../plot/createGlucosePlot';
import { createFoodPLot } from '../plot/createFoodPlot';
import { foodStart } from './keyboards/food';
import { injectionStart } from './keyboards/injection';
import { createInjectionPLot } from '../plot/createInjectionPlot';
import profileWizard from './wizards/profileWizard';
import { profileStart } from './keyboards/profile';
import { userInfoOutput } from './userInfoOutput';
import { calculateDose } from './calculateDose';

export default (bot) => {
  bot.catch((err) => console.error(err));

  const stage = new Scenes.Stage([
    glucoseWizard,
    foodWizard,
    injectionWizard,
    profileWizard,
  ]);

  stage.command('cancel', cancel);

  bot.use(session(), stage.middleware(), upsertUser);

  bot.command('start', (ctx) => ctx.scene.enter(profileWizardId));

  bot.hears(profile, profileStart);
  bot.hears(profileStartIn, (ctx) => ctx.scene.enter(profileWizardId));
  bot.hears(profileStartOut, userInfoOutput);

```

```
bot.hears(glucose, glucoseStart);  
bot.hears(glucoseStartIn, (ctx) => ctx.scene.enter(glucoseWizardId));  
bot.hears(glucoseStartOut, createGlucosePLot);  
  
bot.hears(food, foodStart);  
bot.hears(foodStartIn, (ctx) => ctx.scene.enter(foodWizardId));  
bot.hears(foodStartOut, createFoodPLot);  
  
bot.hears(injection, injectionStart);  
bot.hears(injectionStartIn, (ctx) => ctx.scene.enter(injectionWizardId));  
bot.hears(injectionStartOut, createInjectionPLot);  
  
bot.hears(insulinDose, calculateDose);  
  
bot.hears(toMenu, menuHandler);  
};
```

					ДП 7301.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

src/data/commands/upsertUser.js

```
import { User } from '../models';
```

```
export default async (ctx, next) => {  
  const user = await User.assert(ctx.from.id);  
  ctx.session.user = user;  
  
  return next(ctx);  
};
```

					ДП 7301.00.000 ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

src/data/commands/userInfoOutput.js

```
export const userInfoOutput = (ctx) => {  
  const { height, weight, age, statusDisease } = ctx.session.user;  
  ctx.reply(  
    `Зрiст: ${height}  
    \nBara: ${weight}  
    \nBik: ${age}  
    \nCтан: ${statusDisease}` ,  
  );  
};
```

					ДП 7301.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

src/data/commands/calculateDose.js

```
import User from '../models/User';
import { statusEnum } from '../constants';

const getUser = (userId) => User.findByPk(userId);

const calculateIMT = (m, h) => m / (h / 100) ** 2;

export const calculateDose = async (ctx) => {
  const user = await getUser(ctx.session.user.id);

  let k;
  switch (user.statusDisease) {
    case statusEnum.firstIdentified:
      k = 0.5;
      break;
    case statusEnum.goodCompensation:
      k = 0.6;
      break;
    case statusEnum.volatileCompensation:
      k = 0.7;
      break;
    case statusEnum.decompensation:
      k = 0.8;
      break;
    case statusEnum.ketoacidosis:
      k = 0.9;
      break;
    case statusEnum.pregnancy:
```

					ДП 7301.00.000 ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

```

k = 1;
break;
default:
    return ctx.reply('Некоректно введено статус хворого');
}

const i = calculateIMT(user.weight, user.height);

if (user.age > 13 && user.age < 18) k = 1;

if (i < 18.5) k += 0.1;
if (i > 25) k -= 0.1;

const d = k * user.weight;

const ipd = Math.floor(d / 2);
const morningIPD = 0.6 * ipd;
const eveningIPD = 0.4 * ipd;

const ikd = d - ipd;

return ctx.reply(`Добова потреба: ${d} ОД інсуліну
\nЗ них потреба інсуліну подовженої дії: ${ipd} ОД інсуліну
\nРанкова доза інсуліну подовженої дії: ${morningIPD} ОД інсуліну
\nВечірня доза інсуліну подовженої дії: ${eveningIPD} ОД інсуліну
\nПотреба інсуліну короткої дії: ${ikd} ОД інсуліну
\n\nПодальша корекція дози ІПД - за рівнем глікемії, ІКД - по глікемії і
споживання ХЕ.`);

```

Даний розрахунок - орієнтовний і потребує індивідуальної корекції, що проводиться під контролем рівня глікемії і споживання кількості вуглеводів в ХЕ.);
};

					ДП 7301.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		66

src/data/commands/cancel.js

```
import menuHandler from './keyboards/menu';
```

```
export default async (ctx) => {  
  await ctx.reply('Operation canceled');  
  await ctx.scene.leave();  
  return menuHandler(ctx);  
};
```

					ДП 7301.00.000 ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

src/data/commands/keyboards/food.js

```
import { Markup } from 'telegraf';
```

```
import { foodStartIn, foodStartOut, toMenu } from '../constants';
```

```
export const foodStart = (ctx) =>
```

```
  ctx.reply(
```

```
    'Що хочеш зробити з їжею?',
```

```
    Markup.keyboard([[foodStartIn, foodStartOut], [toMenu]])
```

```
      .oneTime()
```

```
      .resize(),
```

```
  );
```

					ДП 7301.00.000 ПЗ	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дата		

src/data/commands/keyboards/glucose.js

```
import { Markup } from 'telegraf';
```

```
import { glucoseStartIn, glucoseStartOut, toMenu } from '../constants';
```

```
export const glucoseStart = (ctx) =>
```

```
  ctx.reply(
```

```
    'Що хочеш зробити з глюкозою?',
```

```
    Markup.keyboard([[glucoseStartIn, glucoseStartOut], [toMenu]])
```

```
      .oneTime()
```

```
      .resize(),
```

```
  );
```

					ДП 7301.00.000 ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

src/data/commands/keyboards/injection.js

```
import { Markup } from 'telegraf';
```

```
import { injectionStartIn, injectionStartOut, toMenu } from '../constants';
```

```
export const injectionStart = (ctx) =>
```

```
  ctx.reply(
```

```
    "Що хочеш зробити з ін'єкціями?",
```

```
    Markup.keyboard([[injectionStartIn, injectionStartOut], [toMenu]])
```

```
      .oneTime()
```

```
      .resize(),
```

```
  );
```

					ДП 7301.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		70

src/data/commands/keyboards/menu.js

```
import { Markup } from 'telegraf';
```

```
import {
```

```
  glucose,
```

```
  injection,
```

```
  profile,
```

```
  food,
```

```
  insulinDose,
```

```
} from '../constants';
```

```
export default (ctx) =>
```

```
  ctx.reply(
```

```
    'Обери, що хочеш зробити',
```

```
    Markup.keyboard([[glucose, injection], [food, profile], [insulinDose]])
```

```
      .oneTime()
```

```
      .resize(),
```

```
  );
```

					ДП 7301.00.000 ПЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дата		

src/data/commands/keyboards/profile.js

```
import { Markup } from 'telegraf';
```

```
import { profileStartIn, profileStartOut, toMenu } from '../constants';
```

```
export const profileStart = (ctx) =>
```

```
  ctx.reply(
```

```
    'Що хочеш зробити з профілем?',
```

```
    Markup.keyboard([[profileStartIn, profileStartOut], [toMenu]])
```

```
      .oneTime()
```

```
      .resize(),
```

```
  );
```

					ДП 7301.00.000 ПЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

src/data/commands/keyboards/statusDiseaseKeyboard.js

```
import { Markup } from 'telegraf';
```

```
import { statusEnum } from '../constants';
```

```
export const statusDiseaseKeyboard = Markup.keyboard([  
  [statusEnum.firstIdentified, statusEnum.goodCompensation],  
  [statusEnum.volatileCompensation, statusEnum.decompensation],  
  [statusEnum.ketoacidosis, statusEnum.pregnancy],  
])  
  .oneTime()  
  .resize();
```

					ДП 7301.00.000 ПЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

src/data/commands/wizards/foodWizard.js

```
import { Markup, Scenes } from 'telegraf';
import { foodWizardId } from '../constants';
import { EatingData } from '../models';
import menuHandler from '../keyboards/menu';

const skip = 'Пропустити';

const saveCarbohydrates = async (user, { userId, carbohydrates, comment }) => {
  console.log({ user, carbohydrates, comment });
  await EatingData.add(user.id, carbohydrates, comment);
};

const foodWizard = new Scenes.WizardScene(
  foodWizardId,
  (ctx) => {
    ctx.reply('Введіть кількість спожитих вуглеводів (в хлібних одиницях)');

    ctx.scene.session.food = {};

    return ctx.wizard.next();
  },
  (ctx) => {
    const carbohydrates = +ctx.message.text;

    if (ctx.message.text.length < 1 || ctx.message.text.length > 12) {
      return ctx.reply('Недопустима довжина');
    }
  }
);
```

```
ctx.scene.session.food.carbohydrates = carbohydrates;
ctx.reply('Введіть коментар', Markup.keyboard([skip]).oneTime().resize());
return ctx.wizard.next();
},
async (ctx) => {
  if (ctx.message.text !== skip) {
    ctx.scene.session.food.comment = ctx.message.text;
  }

  await saveCarbohydrates(ctx.session.user, ctx.scene.session.food);
  await ctx.reply('Збережено');
  await ctx.scene.leave();
  return menuHandler(ctx);
},
);
export default foodWizard;
```

					ДП 7301.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		75

src/data/commands/wizards/glucoseWizard.js

```
import { Markup, Scenes } from 'telegraf';
import { glucoseWizardId } from '../constants';
import { GlucoseData } from '../models';
import menuHandler from '../keyboards/menu';

const skip = 'Пропустити';

const saveGlucose = async (user, { userId, glucoseLevel, comment }) => {
  console.log({ user, glucoseLevel, comment });
  await GlucoseData.add(user.id, glucoseLevel, comment);
};

const glucoseWizard = new Scenes.WizardScene(
  glucoseWizardId,
  (ctx) => {
    ctx.reply('Введіть результат виміру рівня глюкози (ммоль/л)');

    ctx.scene.session.glucose = {};

    return ctx.wizard.next();
  },
  (ctx) => {
    if (Number.isNaN(+ctx.message.text)) {
      return ctx.reply('Результатом повинно бути числове значення');
    }
    const glucoseLevel = +ctx.message.text;

    ctx.scene.session.glucose.glucoseLevel = glucoseLevel;
```

```
ctx.reply('Введіть коментар', Markup.keyboard([skip]).oneTime().resize());
return ctx.wizard.next();
},
async (ctx) => {
  if (ctx.message.text !== skip) {
    ctx.scene.session.glucose.comment = ctx.message.text;
  }

  await saveGlucose(ctx.session.user, ctx.scene.session.glucose);
  await ctx.reply('Збережено');
  await ctx.scene.leave();
  return menuHandler(ctx);
},
);
export default glucoseWizard;
```

					ДП 7301.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		77

src/data/commands/wizards/injectionWizard.js

```

import { Markup, Scenes } from 'telegraf';
import { injectionWizardId } from '../constants';
import { InjectionData } from '../models';
import menuHandler from '../keyboards/menu';

const skip = 'Пропустити';

const saveInjection = async (user, { userId, injectionDose, comment }) => {
  console.log({ user, injectionDose, comment });
  await InjectionData.add(user.id, injectionDose, comment);
};

const injectionWizard = new Scenes.WizardScene(
  injectionWizardId,
  (ctx) => {
    ctx.reply("Введіть дозу введеної ін'єкції (ОД)");

    ctx.scene.session.injection = {};

    return ctx.wizard.next();
  },
  (ctx) => {
    if (Number.isNaN(+ctx.message.text)) {
      return ctx.reply('Результатом повинно бути числове значення');
    }
    const injectionDose = +ctx.message.text;

    ctx.scene.session.injection.injectionDose = injectionDose;
  }
);

```

```
ctx.reply('Введіть коментар', Markup.keyboard([skip]).oneTime().resize());
return ctx.wizard.next();
},
async (ctx) => {
  if (ctx.message.text !== skip) {
    ctx.scene.session.injection.comment = ctx.message.text;
  }

  await saveInjection(ctx.session.user, ctx.scene.session.injection);
  await ctx.reply('Збережено');
  await ctx.scene.leave();
  return menuHandler(ctx);
},
);
export default injectionWizard;
```

					ДП 7301.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		79

src/data/commands/wizards/profileWizard.js

```

import { Scenes } from 'telegraf';
import { profileWizardId } from '../constants';
import { User } from '../models';
import menuHandler from '../keyboards/menu';
import { statusDiseaseKeyboard } from '../keyboards/statusDiseaseKeyboard';

const updateUser = async (user, { height, weight, age, statusDisease }) => {
  console.log({ user, height, weight, age, statusDisease });
  await User.updateUser(user.id, height, weight, age, statusDisease);
};

const profileWizard = new Scenes.WizardScene(
  profileWizardId,
  (ctx) => {
    ctx.reply('Введіть зріст (в см)');

    ctx.scene.session.user = {};

    return ctx.wizard.next();
  },
  (ctx) => {
    const height = +ctx.message.text;

    if (ctx.message.text.length < 1 || ctx.message.text.length > 12) {
      return ctx.reply('Недопустима довжина');
    }

    ctx.scene.session.user.height = height;
  }
);

```

```

ctx.reply('Введіть вагу (в кг)');
return ctx.wizard.next();
},
(ctx) => {
  const weight = +ctx.message.text;

  if (ctx.message.text.length < 1 || ctx.message.text.length > 12) {
    return ctx.reply('Недопустима довжина');
  }

  // Store the entered name
  ctx.scene.session.user.weight = weight;
  ctx.reply('Введіть вік');
  return ctx.wizard.next();
},
(ctx) => {
  const age = +ctx.message.text;

  if (ctx.message.text.length < 1 || ctx.message.text.length > 12) {
    return ctx.reply('Недопустима довжина');
  }

  ctx.scene.session.user.age = age;
  ctx.reply('Виберіть стан', statusDiseaseKeyboard);
  return ctx.wizard.next();
},
async (ctx) => {
  ctx.scene.session.user.statusDisease = ctx.message.text;

```

```
await updateUser(ctx.session.user, ctx.scene.session.user);  
await ctx.reply('Збережено');  
await ctx.scene.leave();  
return menuHandler(ctx);  
},  
);  
export default profileWizard;
```

					ДП 7301.00.000 ПЗ	Арк.
						82
Змн.	Арк.	№ докум.	Підпис	Дата		

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО”
Кафедра автоматизованих систем обробки інформації та управління

УЗГОДЖЕНО

Керівник проєкту

_____ Ігор Баклан
(підпис) (вл. ім'я, прізвище)

“5” квітня 2021 р.

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Олександр ПАВЛОВ
(підпис) (вл. ім'я, прізвище)

“6” квітня 2021 р.

«Інформаційна система з контролю показників хворих на цукровий
діабет»

ТЕХНІЧНЕ ЗАВДАННЯ

Шифр *ДП 7301.01.000 ТЗ*

на 8 сторінках

Київ – 2021 року

ЗМІСТ

1	ЗАГАЛЬНІ ПОЛОЖЕННЯ	3
1.1	Повне найменування системи та її умовне позначення	3
1.2	Найменування організації-замовника та організацій-учасників робіт	3
1.3	Перелік документів на підставі яких створюється система	3
1.4	Планові терміни початку і закінчення роботи зі створення системи	3
2	ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ	4
2.1	Призначення системи	4
2.2	Цілі створення системи	4
3	ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ	5
4	ВИМОГИ ДО СИСТЕМИ	6
4.1	Вимоги до системи в цілому	6
4.2	Вимоги до функціональних характеристик	6
4.3	Вимоги до видів забезпечення	6
5	СТАДІЇ ТА ЕТАПИ РОЗРОБКИ	7
6	ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ СИСТЕМИ	8

					ДП 7301.01.000 ТЗ			
Зм.	Арк.	Прізвище	Підпис	Дата	Інформаційна система з контролю показників хворих на цукровий діабет	Лім.	Лист	Листів
Розроб.		Валь О.С.						
Перевірів.		Баклан І.В.					2	8
						КПІ ім. Ізгоря Сікорського Каф. АСОІУ Гр. ІС-73		
Н. кон.		Сперкач М.О.						
Затв.		Павлов О.А.						

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Повне найменування системи та її умовне позначення

Повне найменування системи: «Інформаційна система з контролю показників хворих на цукровий діабет».

Умовне позначення: «Diabetes Monitoring System».

Скорочена назва: «DMS».

1.2 Найменування організації-замовника та організацій-учасників робіт

Замовником проекту є кафедра автоматизованих систем обробки інформації і управління факультету інформатики та обчислювальної техніки Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

Представник замовника: Баклан Ігор Всеволодович.

Адреса замовника: м. Київ, пр. Перемоги 37, корп. 18.

Розробником системи є студент групи ІС-73 кафедри Автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» Валь Олег Сергійович.

1.3 Перелік документів на підставі яких створюється система

Завдання на дипломний проєкт є підставою для розробки системи «Інформаційна система з контролю показників хворих на цукровий діабет».

1.4 Планові терміни початку і закінчення роботи зі створення системи

Плановий термін початку робіт над системою – 18.04.2021.

Плановий термін завершення робіт над системою – 04.06.2021.

					ДП 7301.01.000 ТЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ

2.1 Призначення системи

Призначенням розробки є створення телеграм-бота, який дасть змогу хворим на цукровий діабет тримати свої показники, а саме рівень глюкози та інсуліну у нормі. Також забезпечити можливість стежити за своїм раціоном та введеними ін'єкціями інсуліну з будь-яких пристроїв, на яких встановлено телеграм.

2.2 Цілі створення системи

Ціль розробки - допомога хворим на цукровий діабет слідкувати за станом свого здоров'я.

Задачі системи складаються з:

- розробки серверної частини, проектування бази даних;
- реалізації внесення даних до бази;
- реалізації отримання інформації з бази даних;
- реалізація розрахунку потрібної дози інсуліну;

					ДП 7301.01.000 ТЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ

Об'єктом автоматизації є визначення дозування ін'єкцій інсуліну для хворих на цукровий діабет і виводу результатів спостереження за рівнем глюкози, введеними ін'єкціями та спожитими вуглеводами.

					ДП 7301.01.000 ТЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

4 ВИМОГИ ДО СИСТЕМИ

4.1 Вимоги до системи в цілому

Для функціонування системи необхідно мати сервер та клієнт додатку Telegram.

Система повинна оброблювати всі виключні ситуації, що пов'язані з некоректними отриманими даними. Усі дані у системі є конфіденційною інформацією і не можуть бути розголошеними.

Система повинна адекватно реагувати на помилки застосування, видавати відповідні повідомлення користувачі та логувати їх.

4.2 Вимоги до функціональних характеристик

Система повинна виконувати наступні функції:

- перегляд та редагування профілю;
- запис результатів спостереження за рівнем глюкози, введеними ін'єкціями та спожитими вуглеводами до бази даних;
- вивід результатів спостереження;
- розрахунок потрібної дози інсуліну.

4.3 Вимоги до видів забезпечення

Інформаційна система складається з серверу та клієнту додатку Telegram.

Вимоги до серверу:

- ЦП з частотою 1 ГГц чи потужніший;
- ОЗУ об'ємом 512 МБ чи більше;
- 10 ГБ вільного місця на жорсткому диску.

Користувач має мати пристрій, що підтримує додаток Telegram.

					ДП 7301.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

5 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ

У таблиці 5.1 наведено календарний план робіт та терміни їх виконання.

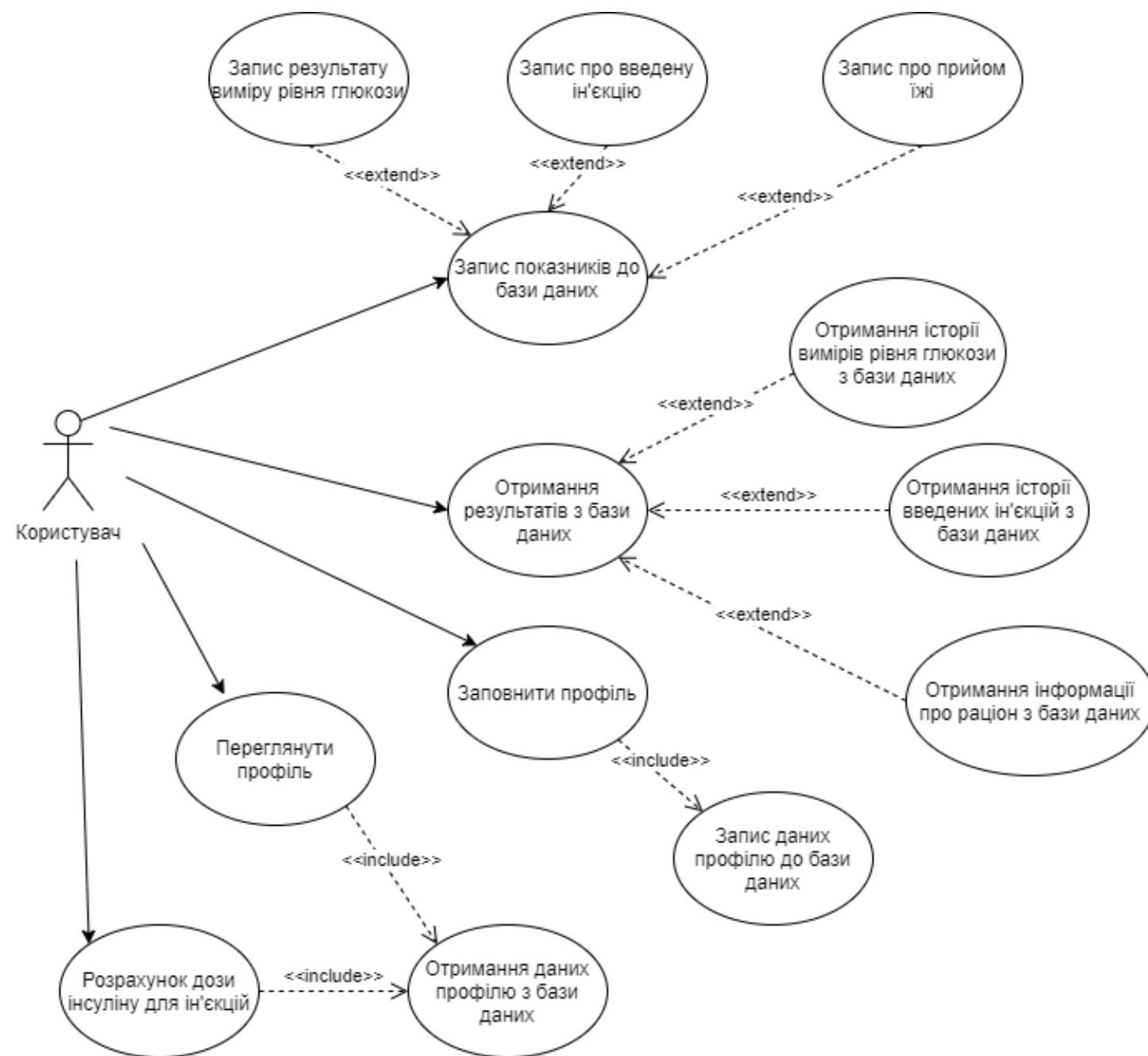
Таблиця 5.1 – Календарний план виконання робіт

<i>№</i>	<i>Назва етапу</i>	<i>Термін виконання</i>
1	Вивчення рекомендованої літератури	12.02.2021
2	Аналіз існуючих аналогів	20.02.2021
3	Постановка та формалізація задачі	03.03.2021
4	Розробка інформаційного забезпечення	16.03.2021
5	Алгоритмізація задач	20.03.2021
6	Обґрунтування використовуваних технічних засобів	29.03.2021
7	Розробка програмного забезпечення	14.04.2021
8	Налагодження програми	22.04.2021
9	Виконання графічних документів	29.04.2021
10	Оформлення пояснювальної записки	04.05.2021
11	Подання ДП на попередній захист	14.05.2021
12	Подання ДП на основний захист	04.06.2021
13	Подання ДП рецензенту	07.06.2021

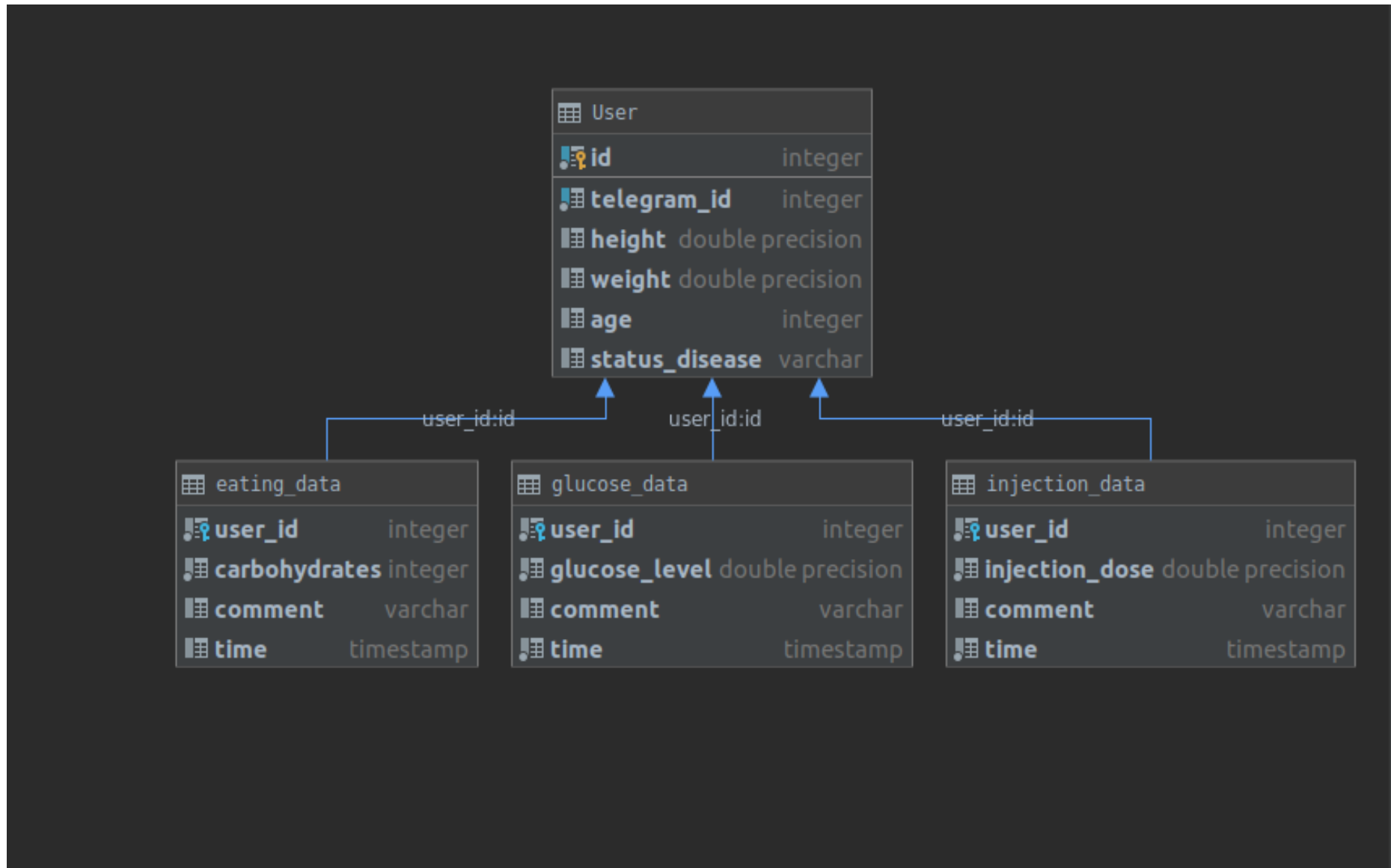
6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ СИСТЕМИ

Для перевірки роботи програмного забезпечення інформаційної системи буде виконано ряд функціональних тестів. Робота цих тестів націлена на перевірку правильності роботи компонентів системи та застосування її в цілому. Також перевіряється виконання передумов тестових сценаріїв, прогонка юніт та е2е тестів.

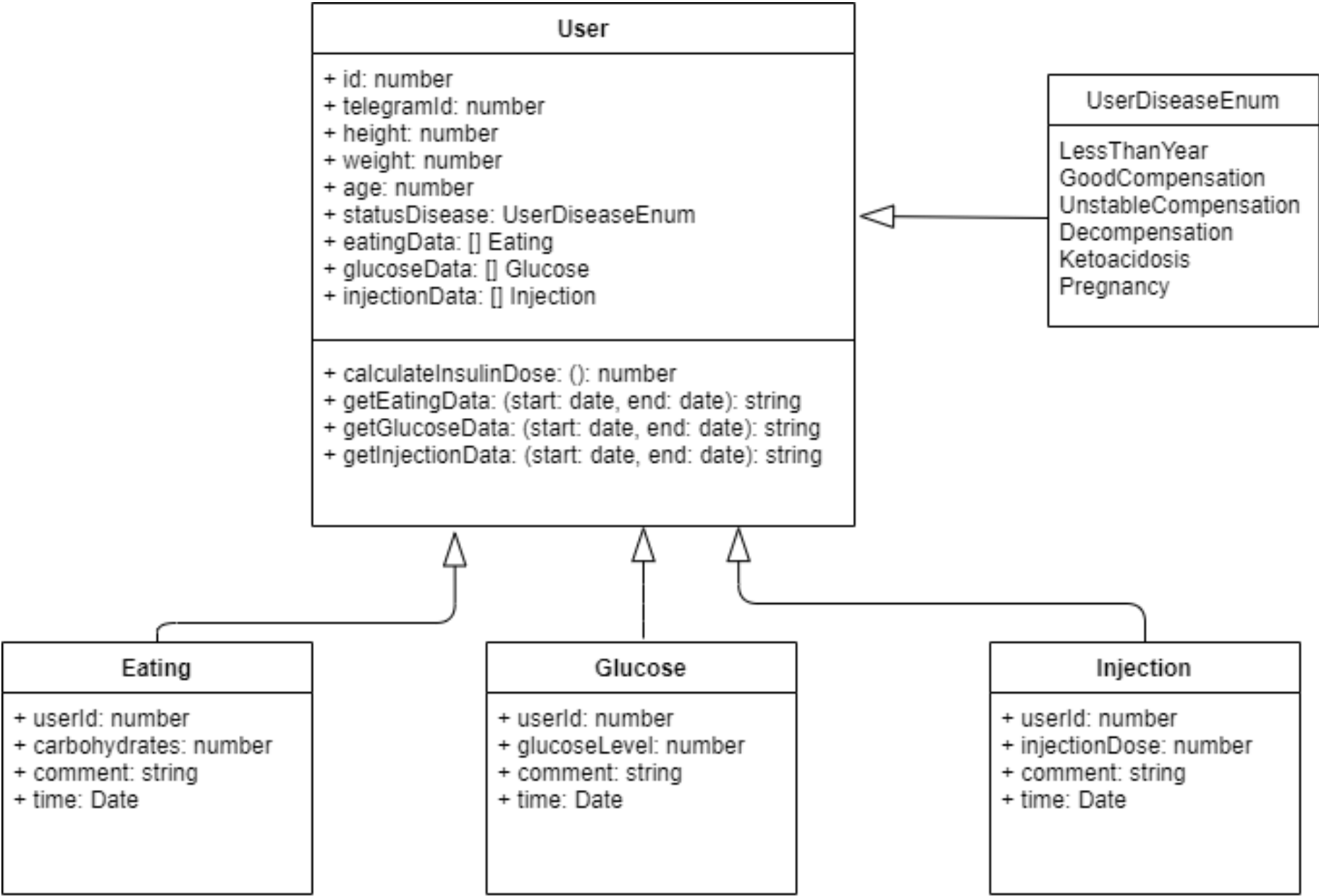
					ДП 7301.01.000 ТЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		



					ДП 7301.02.000 ССВ						
Зм.	Арк.	№ документа	Підпис	Дата	Схема структурна варіантів використання			Літера		Маса	Масштаб
Розробив	Валь О.С.										
Перевірів	Баклан І. В.							Аркуш 1		Аркушів 1	
Консульт.											
Н. кон.	Сперкач М. О.				Інформаційна система з контролю показників хворих на цукровий діабет			КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-73			
Затвердив	Павлов О. А.										



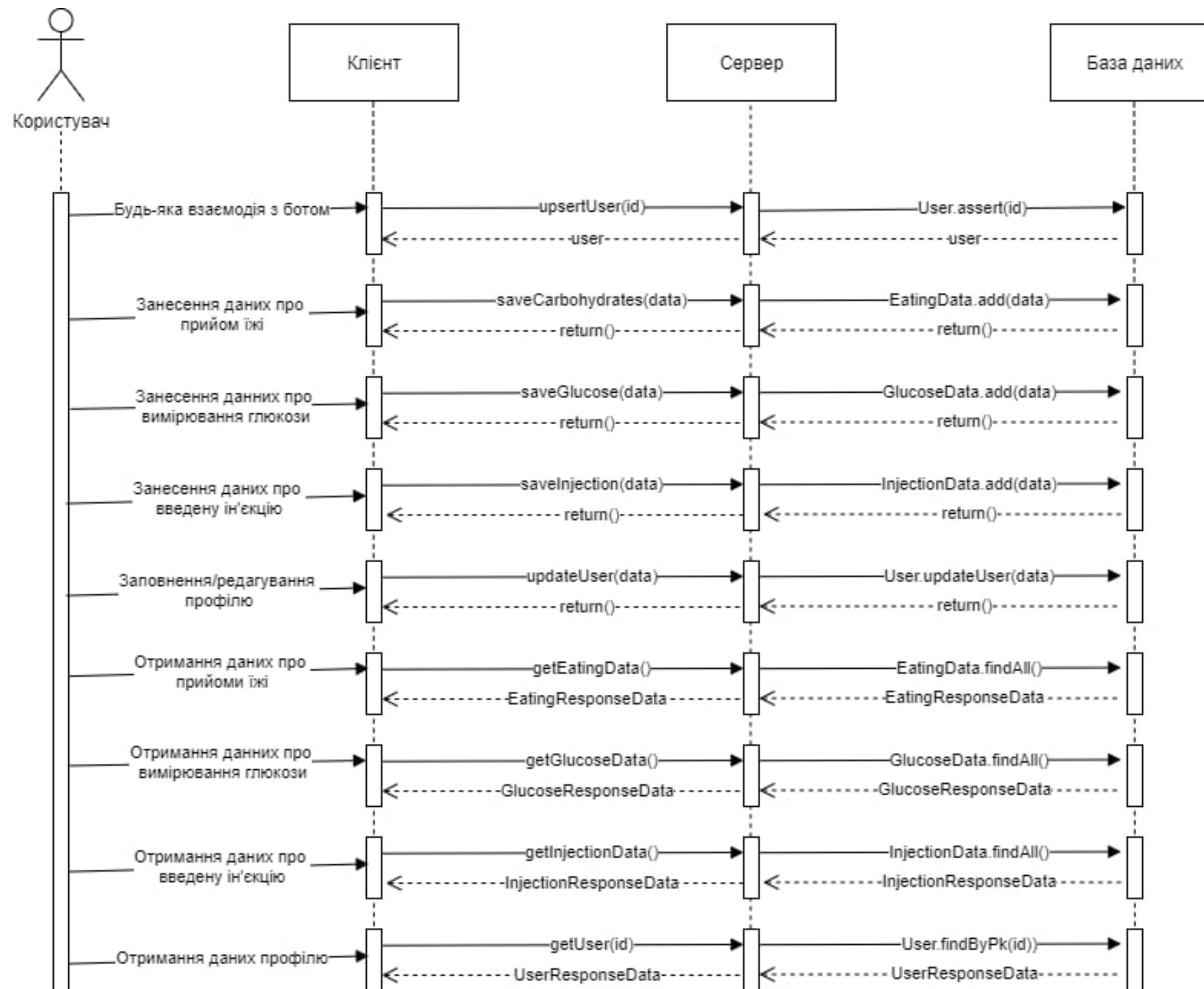
					ДП 7301.03.000 СБД			
					Схема бази даних			
Зм.	Арк.	№ документа	Підпис	Дата				
Розробив		Валь О.С..			Інформаційна система з контролю показників хворих на цукровий діабет			
Перевірів		Баклан І. В.						
Консульт.					КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-73			
Н. кон.		Сперкач М. О.						
Затвердив		Павлов О. А.						



					ДП 7301.04.000 ССКЛ					
					Схема структурна класів програмного забезпечення					
Зм.	Арк.	№ документа	Підпис	Дата						
Розробив		Валь О.С..			Інформаційна система з контролю показників хворих на цукровий діабет					
Перевірив		Баклан І. В.								
Консульт.					КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-73					
Н. кон.		Сперкач М. О.								
Затвердив		Павлов О. А.								



					ДП 7301.05.000 ССК			
					Схема структурна компонентів програмного забезпечення	Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата				
Розробив		Валь О.С.						
Перевірів		Баклан І. В.				Аркуш 1	Аркушів 1	
Консульт.								
Н. кон.		Сперкач М. О.			Інформаційна система з контролю показників хворих на цукровий діабет	КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-73		
Затвердив		Павлов О. А.						



					ДП 7301.06.000 ССП								
					Схема структурна послідовності				Літера		Маса	Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата									
Розробив	Валь О.С.												
Перевірів	Баклан І. В.								Аркуш 1		Аркушів 1		
Консульт.													
Н. кон.	Сперкач М. О.				Інформаційна система з контролю показників хворих на цукровий діабет				КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-73				
Затвердив	Павлов О. А.												

```
calculateDose.js
1 import User from '../models/User';
2 import { statusEnum } from '../constants';
3
4 const getUser = (userId) => User.findById(userId);
5
6 const calculateIMT = (m, h) => m / (h / 100) ** 2;
7
8 export const calculateDose = async (ctx) => {
9   const user = await getUser(ctx.session.user.id);
10
11   let k;
12   switch (user.statusDisease) {
13     case statusEnum.firstIdentified:
14       k = 0.5;
15       break;
16     case statusEnum.goodCompensation:
17       k = 0.6;
18       break;
19     case statusEnum.volatileCompensation:
20       k = 0.7;
21       break;
22     case statusEnum.decompensation:
23       k = 0.8;
24       break;
25     case statusEnum.ketoacidosis:
26       k = 0.9;
27       break;
28     case statusEnum.pregnancy:
29       k = 1;
30       break;
31     default:
32       return ctx.reply('Некоректно введено статус хворого');
33   }
34
35   const i = calculateIMT(user.weight, user.height);
36
37   if (user.age > 13 && user.age < 18) k = 1;
38
39   if (i < 18.5) k += 0.1;
40   if (i > 25) k -= 0.1;
41
42   const d = k * user.weight;
43
44   const ipd = Math.floor(d / 2);
45   const morningIPD = 0.6 * ipd;
46   const eveningIPD = 0.4 * ipd;
47
48   const ikd = d - ipd;
49
50   return ctx.reply(`Добова потреба: ${d} ОД інсуліну
51   \nЗ них потреба інсуліну подовженої дії: ${ipd} ОД інсуліну
52   \nРанкова доза інсуліну подовженої дії: ${morningIPD} ОД інсуліну
53   \nВечірня доза інсуліну подовженої дії: ${eveningIPD} ОД інсуліну
54   \nПотреба інсуліну короткої дії: ${ikd} ОД інсуліну
55   \nПодальша корекція дози ІПД - за рівнем глікемії, ІКД - по глікемії і споживання ХЕ.
56   Даний розрахунок - орієнтовний і потребує індивідуальної корекції, що проводиться під контролем рівня глікемії
57   і споживання кількості вуглеводів в ХЕ.`);
58
59 }
```

					ДП 7301.07.000 СПК						
					Схема прикладу коду для розрахунку дозування інсуліну	Літера		Маса		Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата							
Розробив	Валь О.С.										
Перевірів	Баклан І. В.										
						Аркуш 1		Аркушів 1			
Консульт.					Інформаційна система з контролю показників хворих на цукровий діабет	КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-73					
Н. кон.	Сперкач М. О.										
Затвердив	Павлов О. А.										