

**Гончаров Д.С.**

Чорноморський національний університет імені Петра Могили

**Гончарова Н.В.**

Чорноморський національний університет імені Петра Могили

**Кандиба І.О.**

Чорноморський національний університет імені Петра Могили

## АРХІТЕКТУРА БАГАТОРІВНЕВОЇ СИСТЕМИ МОНІТОРИНГУ СТАНУ ЗДОРОВ'Я ЛЮДИНИ

*У статті досліджено особливості реалізації багаторівневої архітектури системи моніторингу стану здоров'я людини. Проведено аналіз сучасних досліджень в галузі медичних систем, визначено їхнє призначення та основний функціонал. Розглянуто особливості процесу збору даних про фізичний стан людини та особливості реалізації цього процесу. Представлено концептуальну модель багаторівневої архітектури для реалізації функцій системи моніторингу стану здоров'я людини. Наведено опис концепції Internet of Medical Things (IoMT), що використовується в сучасному світі для реалізації функцій відслідковування деяких показників стану здоров'я людини в реальному часі. Визначено основні протоколи передачі даних, що використовуються пристроями концепції IoMT. Наведено схематичну модель передачі даних між застосунками та пристроями IoMT, описано особливості застосування зазначеної концепції. Проаналізовано можливості мови Python в контексті розробки багаторівневої системи моніторингу стану здоров'я людини. Проаналізовано можливість застосування інструментарію Django для створення багаторівневої архітектури інформаційної системи та застосування мови Django Template Language для реалізації рівня представлень системи моніторингу стану здоров'я людини. Проаналізовано можливість візуалізації зібраних даних за допомогою інструментарію Google Chart. Описано можливість застосування засобів Object Relational Mapping для реалізації рівня даних розробленої системи. Визначено найбільш доцільну систему керування базою даних, що дозволяє реалізувати реплікацію даних та підтримує можливість роботи з великою кількістю даних. Наведено опис структури системи моніторингу стану здоров'я людини з використанням балансувача навантаження та кількох окремих серверів для реалізації рівня логіки. Запропоновано шляхи подальшого розвитку багаторівневої системи моніторингу стану здоров'я людини.*

**Ключові слова:** IoMT, Python, Django, багаторівнева архітектура, MySQL.

**Постановка проблеми.** Зростаюча кількість людей, які стикаються з хронічними захворюваннями, такими як діабет, серцево-судинні захворювання, онкологія; обумовлює зростання потреби у спеціалізованих системах моніторингу стану здоров'я людини. Ці системи можуть допомагати відслідковувати та контролювати хронічні захворювання, а також забезпечити швидку реакцію на погіршення стану здоров'я.

Реалізувати систему моніторингу стану здоров'я людини можливо використовуючи різні архітектурні моделі: Monolithic application, Database-centric, Distributed computing, Front end та back end architecture. Проте, використовуючи окремі існуючі архітектури, неможливо в повній мірі реалізувати функції моніторингу стану здоров'я людини:

– модель monolithic application може бути застосована для взаємодії з датчиками, але в ній не передбачена можливість масштабування окремих складових [1, с. 2], наприклад, збору або обробки даних, що є особливо актуальним при роботі системи моніторингу здоров'я людини, що робить цю архітектуру не оптимальною;

– модель Database-centric може використовуватись під час реалізації збору даних, але сама модель не включає елементи балансування навантаження при виконанні значних обсягів запитів [2, с. 3], що виконуватимуться в реалізованій системі моніторингу стану здоров'я;

– модель Front end та back end architecture дозволяє розділити програмне забезпечення на кілька компонентів та знизити навантаження на сервер збору та обробки даних шляхом вико-

ристання обчислювальних ресурсів клієнта, але ця модель не включає в себе можливість балансування навантаження між кількома серверами у разі великої кількості запитів;

– модель Distributed computing призначена для реалізації розподілення навантаження між різними комп'ютерами (серверами), але модель недостатньо уваги приділяє взаємодії розподілених компонентів та баз даних, а також можливості балансування навантаження між розподіленими компонентами.

Таким чином, необхідним є створення системи моніторингу стану здоров'я людини на основі комбінування архітектур програмного забезпечення (ПЗ), що дасть змогу реалізувати централізований збір деяких фізичних показників здоров'я та їхню обробку при наявності великої кількості даних з датчиків різного типу.

**Аналіз останніх досліджень і публікацій.** У дослідженні [3 с. 5] наведено аналіз датчиків моніторингу здоров'я людини, що використовуються в процесі реабілітації. Авторами досліджено різне медичне обладнання для моніторингу стану пацієнтів, що проходять реабілітацію; описано принципи збору та візуалізації даних. В цій роботі також розглянуто можливість інтеграції актуального ПЗ у галузі моніторингу стану здоров'я, з серверами збору даних, наприклад, телемедичні технології у реабілітації пацієнтів з травмами нижніх кінцівок [3 с. 8]. Однак, у роботі не запропоновано варіант реалізації архітектури ПЗ для забезпечення збору та обробки даних з великої кількості датчиків одночасно.

Робота [4 с. 103] містить опис сучасних медичних інформаційних систем. Наведені авторами системи дозволяють організувати взаємодію фахових лікарів з пацієнтами, проводити статистичний аналіз загальної захворюваності, проводити онлайн консультації, збирати результати аналізів тощо. Наведено ґрунтовні дослідження медичних систем, однак недостатньо уваги приділено можливості інтеграції засобів моніторингу стану здоров'я людини та зазначених систем.

Одним з головних напрямів дослідження в галузі медичних інформаційних систем є побудова централізованих систем, наприклад, розробка єдиної медичної інформаційної системи великого міста. У роботі [5 с. 39] наведено опис архітектури інформаційної системи, що дозволить об'єднати лікарні міста та організувати взаємодію між ними. Однак, в роботі недостатньо уваги приділено реалізації можливості зберігання, обробки,

обміну даними моніторингу стану здоров'я людини.

В сучасних дослідженнях увага приділяється можливості застосування мікроконтролерів на зразок Arduino [6 с. 187]. Авторами запропоновано використання шаблону Model-View-Presenter з метою створення ПЗ збору, обробки та відображення даних, але не розглянуто можливість використання спеціалізованих серверів для обробки даних з великої кількості датчиків.

Виконаний аналіз показав, що дослідження системи моніторингу стану здоров'я людини проводяться за різними напрямками: створення медичних інформаційних системи, реалізації централізованих медичних систем великого міста, інтеграція обладнання, побудованого на мікроконтролерах, для відслідковування показників стану здоров'я в реальному часі тощо; але більшість досліджень приділяють недостатньо уваги саме архітектурі зазначених систем, що, в свою чергу, робить питання розробки архітектури розподіленої багаторівневої системи моніторингу стану здоров'я людини актуальним.

**Постановка завдання.** Метою цього дослідження є розробка архітектури та програмного забезпечення для інформаційної технології системи моніторингу стану здоров'я людини. Створення цієї інформаційної системи дозволить забезпечити більш точне та ефективне діагностування захворювань шляхом спрощення процедур збору та аналізу показників фізичного стану людини. Для досягнення зазначеної мети поставлені такі задачі:

- аналіз архітектури та технологій сучасних медичних інформаційних систем;
- розробка багаторівневої архітектури інформаційної системи збору та обробки даних про фізичний стан людини;
- реалізація спроектованої інформаційної системи.

**Виклад основного матеріалу.** Джерелами даних моніторингу стану здоров'я людини можуть бути різні пристрої: мікроконтролери, одноплатні комп'ютери, мобільні пристрої, спеціалізоване медичне обладнання тощо. Реалізувати збір даних з усіх зазначених джерел можливо за допомогою використання технологій вебзапитів, що може бути застосованим на пристроях різної архітектури. Враховуючи, що одночасно до системи може бути підключена велика кількість користувачів та передача даних здійснюватиметься за допомогою вебзапитів, необхідна значна кількість обчислювальних ресурсів.

Швидкий темп розвитку датчиків та медичного обладнання приведе до зростання кількості даних, що надходять до інформаційної системи моніторингу стану людини, а це, в свою чергу, обумовлює потребу масштабування системи під час її використання. Створення зазначеної системи на основі монолітної архітектури неможливе через відсутність можливості масштабування обчислювальних потужностей та функціоналу. Отже, для цієї системи треба використати багаторівневу архітектуру з можливістю балансування навантаження.

Поставлені завдання можна виконати, поділивши систему на рівні, кожен із яких взаємодіє лише з двома сусідніми. Кожен рівень такої архітектури виконує власний обмежений набір функцій. Тому функціональність рівнів можна змінювати без ризику глобальних конфліктів між ними (рис. 1).



Рис. 1. Концептуальне представлення рівнів інформаційної системи моніторингу стану здоров'я людини

Реалізувати збір показників стану людини на фізичному рівні є складною задачею, що в сучасному світі вирішується за допомогою використання концепції Internet of Medical Things (IoMT). Ця концепція полягає в поєднанні різного медичного обладнання та застосунків за допомогою використання мережі Інтернет [7 с. 776].

Концепція IoMT реалізується на базі різних протоколів, але найчастіше для взаємозв'язку обладнання використовується протоколи HTTP та HTTPS. Це, в свою чергу, забезпечує можливість використання різноманітних хмарних техноло-

гій для зберігання та обробки даних та окремий інтерфейс для взаємодії з медичним персоналом. Зазначена особливість дозволяє спростити процес моніторингу медичним персоналом показників стану здоров'я людини завдяки постійній передачі поточних значень [7 с. 776]. Окрім того, інтеграція з різними застосунками забезпечує спрощення доступу до ліків шляхом використання електронних рецептів, автоматизацію ведення історії хвороби при взаємодії з сервісами електронних карток пацієнтів (рис. 2).

Можна інтегрувати багаторівневу архітектуру та концепцію IoMT, використовуючи сучасні мови програмування та спеціалізовані фреймворки для обміну даними за допомогою протоколів HTTP та HTTPS.

Мова програмування загального призначення Python є однією з найбільш розповсюджених та сучасних мов. Регулярне оновлення та вдосконалення Python забезпечує підтримку сучасних концепцій, підходів до програмування. Окрім того, наявність відкритого репозиторію з великою кількістю різних бібліотек дозволяє легко масштабувати функціонал розроблених застосунків.

Python підтримує широкий спектр інструментарію для аналізу даних та реалізації різних технологій, у тому числі вебсервери для обробки запитів HTTP та HTTPS [8 с. 5]. Найбільш розповсюдженим та функціональним засобом реалізації вебсерверу для Python є фреймворк Django.

Django є високорівневим вебфреймворком Python, що дозволяє створення застосунків з архітектурою Model-View-Controller (MVC) [9 с. 23]. Використання цього фреймворку дає змогу розділити рівні подання, логіки та даних за допомогою патерну MVC.

Рівень подання у багаторівневій системі моніторингу стану здоров'я людини на базі Django може бути представлений у вигляді HTML сторінки, що генерується за допомогою шаблону, описаного мовою Django Template Language (DTL). DTL дозволяє динамічно додавати в статичний шаблон HTML контент з можливістю використання мови JavaScript (JS) та додаткових бібліотек для нього.

Візуальна інформація може бути ще більш наглядною та легше сприйматись звичайними користувачами (пацієнтами), аніж інформація іншого типу. Окрім того, аналізуючи графічне зображення, значно простіше виявити відхилення у зібраних показниках. Отже, для повноцінної реалізації рівня подання в системі моніторингу стану здоров'я необхідна інтеграція інструментарію візуалізації даних.

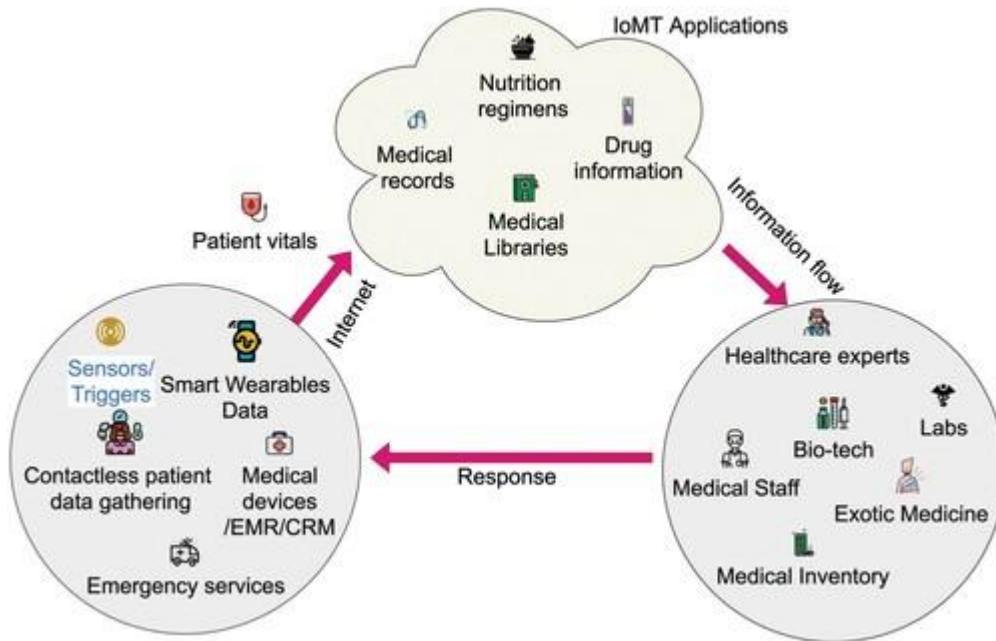


Рис. 2. Схематична модель використання ІоМТ [7 с. 777]

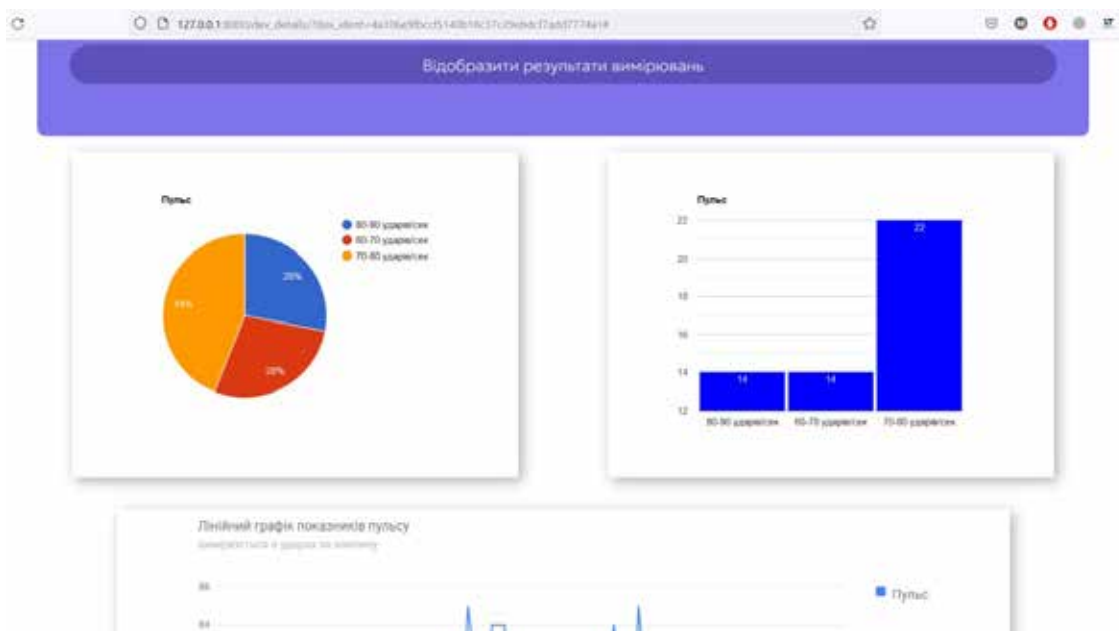


Рис. 3. Графічне представлення зібраних даних

Задіяний на рівні представлення JS дозволяє візуалізувати дані у вигляді гістограм, кругових та кореляційних діаграм тощо. Графічне відображення зазначених діаграм забезпечується інструментарієм Google Chart. Цей інструментарій, представлений у вигляді окремої бібліотеки JS, є безкоштовним та може бути інтегрований з будь-яким HTML шаблоном.

В багаторівневій системі моніторингу стану здоров'я людини Google Chart дозволяє створити графічне відображення зібраних показників окси-

генації, пульсу, кров'яного тиску тощо. Використання DTL у поєднанні з JS дозволяє здійснити параметризовану вибірку необхідних вимірювань та їх динамічне відображення (рис. 3).

Рівень логіки застосунків, побудованих на основі фрейворку Django, реалізується за допомогою мови Python. Взаємозв'язок коду Python та шаблонів DTL здійснюється за допомогою Django та засобами GET та POST запитів. Такими чином, на рівні логіки можна інтегрувати засоби обробки даних, що підтримуються мовою програмування

Python, включно з інструментарієм штучного інтелекту, який є актуальним напрямом в сучасній медицині.

Нормальне функціонування системи моніторингу стану здоров'я людини вимагає реалізації на рівні логіки коректної обробки отриманих запитів та передачу показників до рівня даних. Враховуючи, що система має підтримувати сучасну концепцію IoT та забезпечувати безпеку даних користувача, використана вбудована технологія CSRF protection, що унеможливує доступ до даних неавторизованих користувачів. При цьому для відображення отриманих зібраних даних користувачу використано інструментарій авторизації Django, що також запобігає неавторизованому доступу до них.

Отримані дані мають бути збережені та доступні за запитом користувача, що, в свою чергу, забезпечується засобами Object Relational Mapping (ORM) фреймворку Django. Використання ORM значно спрощує розробку коду необхідного для коректної взаємодії з системою керування базою даних (СКБД).

Інтеграція зазначених технологій захисту даних вимагає наявності спеціалізованих таблиць в базі даних (БД). Мінімальний функціонал системи моніторингу стану здоров'я людини має включати в себе відслідковування кров'яного тиску, оксигенації, частоти серцебиття (пульсу), температури та даних користувача для реалізації концепції IoT. Враховуючи перелічені показники, вимоги

безпеки та дані інструментарію логування Django, структура БД містить 18 таблиць (рис. 4).

За замовчування проекти створенні на основі Django використовують у якості СКБД SQLite, але ця СКБД має ряд недоліків, що не дозволяють її використовувати для роботи з даними в системі моніторингу стану здоров'я людини. Основним недоліком SQLite є відсутність спеціалізованого серверу, що унеможливує реалізацію реплікації та кешування даних для пришвидшення доступу.

Доцільнішим є використання реляційної СКБД MySQL, яка працює на окремому сервері та може бути реплікована вбудованими засобами, а наявний інструментарій Query Cache забезпечує швидкодію при роботі з великими об'ємами даних. Зазначені переваги у сукупності з вбудованими драйверами Django роблять цю СКБД оптимальним рішенням.

Велика кількість запитів до БД в системі моніторингу стану здоров'я людини може занадто навантажувати сервер СКБД та сповільнювати його роботу. Доцільним рішенням є використання кількох паралельно працюючих серверів для реалізації рівня логіки та кількох серверів СКБД з реалізацією реплікації для зменшення навантаження.

Підвищити надійність роботи представленої системи можливо шляхом підключення додаткових вебсерверів та додавання балансувача навантаження. Отже, ефективна робота системи моніторингу стану здоров'я людини має містити три

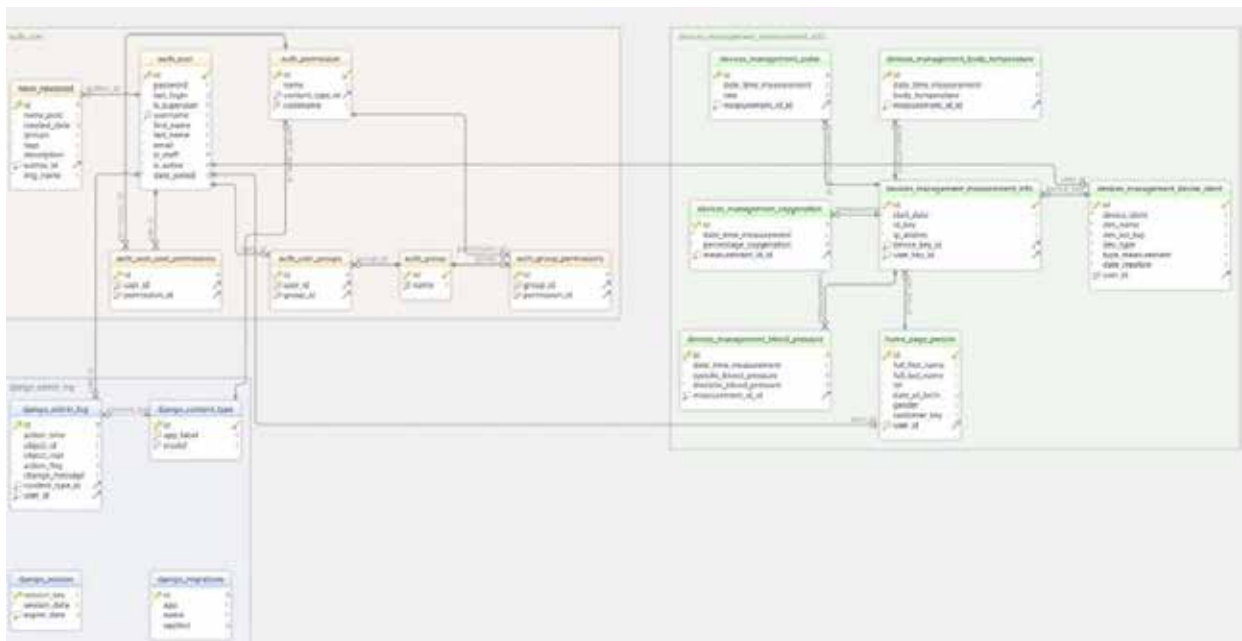


Рис. 4. Структура БД наведеної системи моніторингу стану здоров'я людини

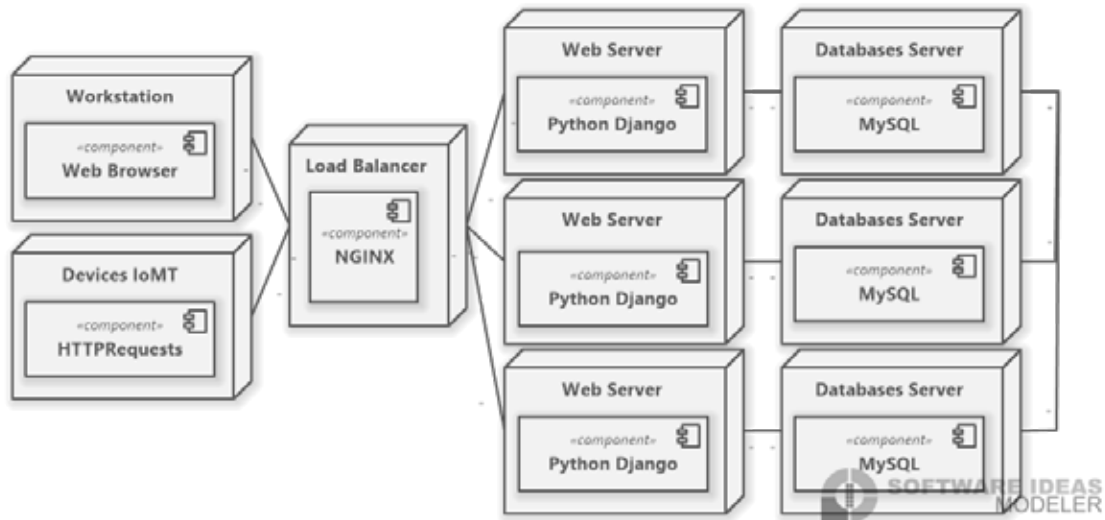


Рис. 5. Структура системи моніторингу стану здоров'я людини

основні компоненти: балансувач навантаження, вебсервери, що можуть працювати паралельно для обробки великої кількості запитів та кілька серверів БД (рис. 5).

**Висновки.** Виконано дослідження сучасних концепцій, на основі яких будуються системи моніторингу стану здоров'я. Представлено багаторівневу архітектуру для побудови зазначених

систем з підтримкою можливості паралельної обробки запитів. Розроблено інформаційну технологію для збору, зберігання та відображення даних стану здоров'я людини на основі мови програмування Python та фреймворку Django.

У подальшому планується інтеграція до розробленої системи засобів статистичного аналізу та методів ШІ для обробки отриманих даних.

#### Список літератури:

- 1) Ren Z., Wang W., Wu G. Migrating web applications from monolithic structure to microservices architecture. Proceedings of the 10th Asia-Pacific Symposium on Internetware(2018). P. 1–10.
- 2) Vicente A., Etcheverry L., Sabiguero A. An RDBMS-only architecture for web applications. 2021 XLVII Latin American Computing Conference (CLEI)(2021). P. 1–9.
- 3) Марценюк В. П., Качур І. В., Сверстюк А. С. Моніторинг стану здоров'я за функціональними показниками за допомогою сенсорів у реабілітаційній медицині: систематичний огляд. Вісник наукових досліджень. Том 2. С. 5–12.
- 4) Лирчиков В. О., Байбуз О. Г. ОГЛЯД СУЧАСНИХ МЕДИЧНИХ ІНФОРМАЦІЙНИХ СИСТЕМ. Актуальні проблеми автоматизації та інформаційних технологій. Том. 25, 2021. С. 103–107.
- 5) Хвищун А. І., Качмар В. О. Принципи формування єдиної медичної інформаційної системи великого міста. Medical Informatics and Engineering. Том 3. С. 39–47.
- 6) Шевченко К. Л., Склярєвський А. О. Принципи побудови систем моніторингу стану здоров'я людини. Мехатронні системи: інновації та інжиніринг(2021). С. 186–187.
- 7) Razdan S., Sharma S. Internet of medical things (IoMT): Overview, emerging technologies, and case studies. IETE technical review. Vol. 39, Issue 4. P. 775–788.
- 8) Ziogas A. N., Schneider T., Ben-Nun T. Productivity, portability, performance: data-centric Python. Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis(2021). P. 1–13.
- 9) Mele A. Django 3 By Example: Build powerful and reliable Python web applications from scratch. Packt Publishing Ltd, 2020. 533 p.

#### **Honcharov D.S., Honcharova N.V., Kandyba I.O. ARCHITECTURE OF A MULTI-LEVEL HUMAN HEALTH MONITORING SYSTEM**

*The article studies the peculiarities of multi-level architecture implementation of a system for human health state monitoring. Thus, modern researches of medical systems are analysed, the purpose and basic functionality of these systems are determined. The peculiarities of the collecting data process for monitoring the human condition and the features of implementing this process are considered. Moreover, a conceptual model of a*

*multi-level architecture for implementing the functions of a human health monitoring system is presented. A description of the Internet of Medical Things (IoMT) concept used in the modern world to implement the functions of human health monitoring is given. In addition, the main data transmission protocols used by IoMT devices are defined. A schematic model of data transfer between applications and IoMT devices is presented, and the features of the application of this concept are described. The capabilities of the Python language in the context of developing a multi-level human health monitoring system are analysed. The possibility of using Django tools to create a multi-level architecture of an information system is considered. Thus, the paper includes a description of the possibility of using the Django Template Language to implement the human health monitoring system presentation level. The possibility of visualising the collected data using the Google Chart toolkit is analysed. Also, the opportunity of using Object Relational Mapping tools to implement the data layer of the developed system is described. The most appropriate database management system is determined, which allows to implement data replication and supports the ability to work with a large amount of data. A description of the structure of the human health monitoring system using a load balancer and several separate servers to implement the logic level is given. The ways of further development of a multilevel system for monitoring the state of human health are proposed.*

**Key words:** *IoMT, Python, Django, multi-level architecture, MySQL.*