

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського  
«Харківський авіаційний інститут»

Факультет програмної інженерії та бізнесу

Кафедра інженерії програмного забезпечення

## Пояснювальна записка до дипломного проєкту

магістра

(освітній ступінь)

на тему «Дослідження моделей та методів керування системою  
розумного ліжка»

XAI.603.6-95пз1.121.167331.200

Виконав: студент 6 курсу групи № 6-95пз1  
Спеціальність 121 – Інженерія програмного  
забезпечення

(код та найменування)

Освітня програма Хмарні обчислення та  
Інтернет речей

(найменування)

Тягнирядно А.В.

(прізвище й ініціали студента)

Керівник Туркін І.Б.

(прізвище та ініціали)

Рецензент Ткачов В.М.

(прізвище та ініціали)

Харків – 2020

**Міністерство світи і науки України**  
**Національний аерокосмічний університет ім. М. Є. Жуковського**  
**«Харківський авіаційний інститут»**

Факультет програмної інженерії та бізнесу

(повне найменування)

Кафедра інженерії програмного забезпечення

(повне найменування)

Рівень вищої освіти другий (магістерський)

Спеціальність 121 – інженерія програмного забезпечення

(код та найменування)

Освітня програма хмарні обчислення та Інтернет речей

(найменування)

**ЗАТВЕРДЖУЮ**

**Завідувач кафедри**

І. Б. Туркін

(підпис)

(ініціали та прізвище)

“ ”

2020 року

**З А В Д А Н Н Я**  
**НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ**

Тягнирядно Андрій Валерійович

(прізвище, ім'я, по батькові)

1. Тема дипломного проекту Дослідження моделей та методів керування системою розумного ліжка

керівник дипломного проекту Туркін Ігор Борисович, д.т.н., професор

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

затверджені наказом вищого навчального закладу від “ ” 2020 року №

2. Термін подання студентом роботи

3. Вихідні дані до роботи: моделі та методи керування системою розумного ліжка

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

провести аналіз та моделювання проблемної області керування розумними ліжками; дослідити моделі надання знань; розробити математичну модель для керування розумним ліжком; створити базу даних для зберігання даних з розробленої моделі; спроектувати архітектуру системи; розробити алгоритми придбання знань та висновку для керування моделлю розумного ліжка; програмно реалізувати систему

5. Перелік графічного матеріалу

РПЗ – стор. 73, рисунків – 24 шт., таблиць – 5 шт., презентація – 16 слайдів.

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Туркін І.Б., зав. каф. 603		
2	Туркін І.Б., зав. каф. 603		
3	Туркін І.Б., зав. каф. 603		

8. Нормоконтроль \_\_\_\_\_ В.А. Постернакова « \_\_\_\_ » \_\_\_\_\_ 2020 р.  
(підпис) (ініціали та прізвище)

7. Дата видачі завдання \_\_\_\_\_

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту	Строк виконання етапів проекту	Примітка
1	Отримання і затвердження теми диплому	03.09.2019	
2	Аналіз предметної області	04.09.2019	
3	Постановка задачі	20.11.2019	
4	Проведення теоретичних досліджень	22.11.2019	
5	Розробка прототипу ПЗ	02.09.2020	
6	Підготовка пояснювальної записки	22.10.2020	
7	Оформлення пояснювальної записки до дипломного проекту	10.11.2020	
8	Передзахист дипломного проекту	27.11.2020	
9	Захист дипломного проекту	24.12.2020	

Студент

\_\_\_\_\_  
( підпис )

Тягнирядно А.В.  
(прізвище та ініціали)

Керівник роботи

\_\_\_\_\_  
( підпис )

Туркін І.Б.  
(прізвище та ініціали)

## РЕФЕРАТ

Пояснювальна записка до дипломного проєкту містить 73 стор., 24 рис., 19 джерел.

Об'єкт дослідження - процес керування «Розумним ліжком».

Предмет дослідження – моделі та методи керування «Розумним ліжком».

Метою дослідження є підвищення ефективності системи керування «Розумним ліжком» за рахунок розробки програмної системи розумного ліжка.

Для досягнення поставленої мети необхідно вирішити ряд завдань: провести аналіз та моделювання проблемної області керування розумними ліжками; дослідити моделі надання знань; розробити математичну модель для керування розумним ліжком; створити базу даних для зберігання даних з розробленої моделі; спроектувати архітектуру системи; розробити алгоритми придбання знань та висновку для керування моделлю розумного ліжка; програмно реалізувати систему.

Наукова новизна. Вперше розроблено модель керування «Розумним ліжком», яка на відміну від існуючих використовує продукційні моделі в якості моделей надання знань, що надасть змогу обирати оптимальне положення «розумного ліжка» для сну.

Практична значимість отриманих результатів. В результаті досліджень була розроблена програмна система «Розумне ліжко» яка дозволить забезпечити користувачеві правильний і повноцінний сон за рахунок вибору оптимального стану ліжка.

ІНТЕРНЕТ РЕЧЕЙ, РОЗУМНЕ ЛІЖКО, ШТУЧНИЙ ІНТЕЛЕКТ,  
МОДЕЛЬ НАДАННЯ ЗНАНЬ, ПРОДУКЦІЙНА МОДЕЛЬ

## **ABSTRACT**

Explanatory note to the master's thesis 73 pp., 24 fig., 19 sources.

The object of study - the process of managing the "Smart Bed".

The subject of research - models and methods of control "Smart Bed".

The purpose of the study is to increase the efficiency of the control system "Smart Bed" by developing a software system for a smart bed.

To achieve this goal it is necessary to solve a number of tasks: to analyze and model the problem area of smart bed management; explore models of knowledge delivery; develop a mathematical model to control a smart bed; create a database to store data from the developed model; design system architecture; develop algorithms for acquiring knowledge and conclusions to control the model of a smart bed; software to implement the system.

Scientific novelty. For the first time, a Smart Bed control model has been developed, which, unlike the existing ones, uses production models as knowledge models, which will allow to choose the optimal position of the "smart bed" for sleep.

The practical significance of the results obtained. As a result of research, a software system "Smart Bed" was developed, which will provide the user with proper and complete sleep by choosing the optimal condition of the bed.

**INTERNET OF THINGS, SMART BED, ARTIFICIAL INTELLIGENCE,  
KNOWLEDGE MODEL, PRODUCTION MODEL**

## ЗМІСТ

Вступ.....	5
1 Аналіз проблемної галузі та постановка задачі.....	7
1.1 Аналіз предметної галузі.....	7
1.2 Аналіз існуючих аналогів.....	10
1.3 Постановка задачі.....	13
2 Математична модель для керування розумним ліжком .....	16
2.1 Аналіз та моделювання предметної області керування сном.....	16
2.2 Розробка моделі розумного ліжка.....	22
2.3 Дослідження моделей надання знань.....	24
2.5 Розробка моделі для керування розумним ліжком .....	34
2.6 Розробка продукційної моделі для керування ліжком.....	40
2.7 Розробка бази даних.....	45
2.8 Розробка алгоритмів придбання знань та висновку для керування моделлю розумного ліжка.....	48
3 Розробка програмного продукту «Розумне ліжко».....	53
3.1 Розробка архітектури системи.....	53
3.2 Опис обраних технологій.....	55
3.3 Опис фізичної моделі бази даних.....	57
3.4 Інтерфейс і функціонал.....	59
Висновки.....	70
Перелік джерел посилання.....	72

## ВСТУП

Сьогодні інтернет речей (internet of things, IoT) стає все більш популярним: з'являється багато розумних будинків, транспортів і підприємств. Популярність інтернету речей можна коротко описати як збільшення кількості пристроїв, які взаємодіють з мережею інтернет та мережею інших таких пристроїв, а не тільки з користувачем, як було раніше. Поява і розвиток в світі інтернету речей були очікувані, адже в сучасному світі з плином часу все більшу кількість проблем ми перекладаємо на машини і роботи для швидкого і ефективного їх вирішення.

Повернемося до основної мети появи інтернету речей і дуже швидкого їхнього розвитку – спрощувати людям життя. І дійсно: в сучасному світі темп життя зростає з кожним днем, набагато збільшується кількість стресу, це призводить до накопичення втоми, і, як наслідок, людина не в змозі підтримувати постійну активність життя, різко падає ефективність його діяльності. Одним із способів підтримки людини є, наприклад, повноцінний правильний відпочинок. Людина проводить в стані сну близько третини свого життя. Ми звикли вважати сон періодом спокою та бездіяльності, проте сон – вкрай важливий стан, що впливає на всі сторони повсякденного життя. Від нього залежать здоров'я, настрої, емоції, поведінка, професійна діяльність людини. Коли ми спимо, в організмі відбувається відновлення витрачених енергетичних ресурсів.

Правильний сон допоможе людині підтримувати ритм свого життя, не дозволяти втоми впливати на неї, але як можна висипатися краще за одне і те ж час? Іншими словами, як зробити сон більш ефективним? Відповідь практично очевидний: використовувати правильне ліжко.

Всі ми періодично мучимося ночами, не в змозі заснути: ліжко може бути занадто жорстке або, навпаки, м'яке, чи нам дуже холодно навіть під ковдрою, подушка занадто нагрілася, а заповітна прохолодна сторона абсолютно не тримає потрібну нам форму, і голова кудись то провалюється... та й коли прокидаємося,

навіть якщо дотримуємося режиму, можемо кожного ранку відчувати себе по-різному: один раз ми сповнені енергії і готові вершити великі справи, а іншого дня ми ледве знаходимо в собі сили не те, що встати, а навіть відкрити очі. Кожна людина хоча б кілька разів так стикалася з однією з цих проблем.

Саме для уникнення таких ситуацій буде розроблена система розумного ліжка. Кожна з перерахованих вище проблем буде певним чином вирішуватися, дозволяючи користувачеві не тільки зручно налаштувати своє ліжко так, як підходить йому самому, але і вибрати певні налаштування для найбільш вигідного виду відпочинку.

Об'єкт дослідження - процес керування «Розумним ліжком».

Предмет дослідження – моделі та методи керування «Розумним ліжком».

Метою дослідження є підвищення ефективності системи керування «Розумним ліжком» за рахунок розробки програмної системи розумного ліжка.

Для досягнення поставленої мети необхідно вирішити ряд завдань: провести аналіз та моделювання проблемної області керування розумними ліжками; дослідити моделі надання знань; розробити математичну модель для керування розумним ліжком; створити базу даних для зберігання даних з розробленої моделі; спроектувати архітектуру системи; розробити алгоритми придбання знань та висновку для керування моделлю розумного ліжка; програмно реалізувати систему.

Наукова новизна. Вперше розроблено модель керування «Розумним ліжком», яка на відміну від існуючих використовує продукційні моделі в якості моделей надання знань, що надасть змогу обирати оптимальне положення «розумного ліжка» для сну.

Практична значимість отриманих результатів. В результаті досліджень була розроблена програмна система «Розумне ліжко» яка дозволить забезпечити користувачеві правильний і повноцінний сон за рахунок вибору оптимального стану ліжка.



# 1 АНАЛІЗ ПРОБЛЕМНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

## 1.1 Аналіз проблемної галузі

Сон – це найважливіший фізіологічний процес, який необхідний для нормального функціонування органів людини. За сучасними уявленнями під час сну центральна нервова система перемикається на обробку інформації, що надходить від внутрішніх органів. Отже, будь-яке порушення сну буде негативно позначатися на стані здоров'я людини. В умовах індустріального та пост-індустріального суспільства високий рівень стресу тісно пов'язаний з порушеннями сну. Світлова зашумленість простору, в тому числі виділеного для сну, постійний шум і вібрації міст негативно позначаються на здоров'ї і якості сну, а рівень стресу сучасної людини передбачає необхідність якісного і правильного відпочинку, тобто сну. З точки зору психології цей же соціально-економічний контекст з високим рівнем стресу поступово виснажує ресурси організму. Для того, щоб нормалізувати функцію сну корисні не тільки сучасні медикаменти, але і правильне положення під час сну.

Останнім часом наші біологічні ритми страждають через те, що праця стає все більш напруженою, відпочинок коротшим, а життя навколо все більш гучнішим. Генетично запрограмовані ритми фізіологічних функцій порушуються, і поступово погіршується самопочуття, знижуються творчі та фізичні сили, формуються хвороби [1]. Щороку навесні і восени ми стаємо заручниками переходу на літній і зимовий час, що зменшує витрату електроенергії, але ламає усталені біоритми і завдає непоправної шкоди здоров'ю.

Тому досить актуальною стає тема оптимізації та автоматизації процесу налагодження здорового сну, як і діджиталізація інших галузей життя людини. Це обумовлює такий розвиток і популярність IoT (internet of things, інтернет речей), який зараз називають одним із найголовніших трендів мирової економіки в найближче десятиріччя. Можливість отримувати значну кількість інформації, яка

згодом буде опрацьована інструментами аналізу даних, та ефективно прийняття рішень на основі результатів цього аналізу робить технології і предмети інтернету речей надзвичайно затребуваними в наших реаліях. [2]

Варто зауважити, що без спеціальних алгоритмів оптимізації будь-яких процесів предмети інтернет речей можуть називатися “розумними” тільки умовно завдяки зв’язок з мережею. Насправді ж реально розумними предмети інтернету речей роблять механізми штучного інтелекту, які дозволяють не тільки виконувати окремі команди користувачів, але й приймати власні рішення на основі інформації, що надходить, для оптимізації профільного процесу. Якщо раніше людина обмежувалася використанням роботів для виконання важкої фізичної роботи, то зараз із кроком науки вперед ми маємо можливість вирішувати більш складні, в тому числі складні із математичної точки зору, задачі саме завдяки штучному інтелекту.

Поєднавши дві галузі – сну та штучного інтелекту – отримаємо предмет інтернету речей розумне ліжко. Саме ліжко може впливати на багато ключових параметрів, що впливають на якість сну людини, які будуть розглянуті в ході роботи. Але системи ліжок, що виконують команди користувача вже існують, що буде розглядатися надалі в наступних розділах, нам же необхідно створити систему, що буде здатна самостійно обирати, в який стан перейти, щоб досягнути оптимальної якості сну користувача. Тож маємо додатково розглянути, як саме системи штучного інтелекту можуть приймати рішення.

Щоб перейти від розрізненої інформації предметної галузі до структурованого вирішення проблем системами штучного інтелекту, необхідно використовувати деяку модель надання знань. Суть в тому, що знання, якими володіє людина, має бути структурована для використання програмними системами, моделі надання знань додають саме систематизації до розрізнених знань.

Надзвичайно важливо використовувати саме ту модель надання знань, що підходить до предметної галузі та можливостей системи, що плануються до

розробки. Існує декілька найбільш популярних та досліджених моделей, усі зі своїми перевагами та недоліками, що надасть можливість обрати саме ту, яку доцільно буде використовувати.

Є багато факторів, що можуть впливати на якість сну людини, і, щоб розробити ефективну систему впливу на неї, необхідно виокремити основні з них та використати в системі, що розробляється. Вкрай важливо дослідити, як ці параметри співпрацюють один із іншим, якими можна оцінювати якість сну, які входять у передумови впливу на сон, а які можна використовувати саме в системі предмету інтернет речей. Моделювання предметної галузі керування сном надасть саме розуміння зв'язків між цими параметрами та наочно представить, як можна використовувати їх в програмних системах. Тільки після повного моделювання можна буде скласти систему моделі надання знань для ефективного використання цих знань.

Важливо також зауважити, що не завжди можна ефективно використовувати стандартні алгоритми роботи із моделями знань в таких системах. Тому доцільно буде оптимізувати деякі з них, щоб підвищити ефективність системи, що розробляється.

Використовуючи алгоритми штучного інтелекту та найбільш відповідні предметній області моделі надання знань, система розумного ліжка зможе аналізувати весь спектр інформаційних параметрів, що надходять від користувача, та правильно обрати оптимальне положення людини під час сну.

Однак складена система моделі надання знань дозволяє лише алгоритмічно обробляти дані, якими володіє система. Для зберігання цієї інформації необхідно спроектувати базу даних таким чином, що з неї можна ефективно отримувати будь-яку інформацію за найкоротші строки. Щоб зберегти баланс між простотою структури бази даних та швидкістю отримання інформації з неї, необхідно додатково нормалізувати базу. Нормалізація потрібна, щоб збільшити структурованість інформації та уникнути її дублювання, як наслідок, це допоможе

запобігти великій кількості помилок та додаткових операцій, щоб привести систему до правильного стану.

Також слід зазначити, що розробка системи з галузі інтернету речей потребує проектування дещо специфічної архітектури. По перше, як і в будь-якому великому проекті, для спрощення розуміння та подальшої розробки системи архітектура повинна бути багат шаровою. Це робить систему набагато більше придатною до модифікацій, впровадження нового функціоналу та ін. По друге, розробка системи інтернету речей потребує розробки механізму передачі інформації між сервером, де виконується основна логіка та отримується інформація, та самим предметом інтернету речей. Оптимально, якщо окремі шари для такого міжмодульного спілкування будуть існувати і на сервері, і безпосередньо на процесорі предмету інтернету речей або емуляторі.

Тож у даній роботі буде розглядатися вплив якості сну на якість життя в умовах все зростаючого її темпу, вплив різних параметрів сну на якість сну, а також варіанти оптимізації впливу на сон із метою поліпшити якість життя людини. Буде розроблена система розумного ліжка “Beddest” та обладнана алгоритмами штучного інтелекту із відповідними моделями надання знань для самостійного прийняття рішення про оптимізацію сну користувача.

## 1.2 Аналіз існуючих аналогів

Щоб сформулювати перелік вимог до кінцевого продукту необхідно проаналізувати деякі аналоги, що існують. У цьому розділі роздивимось приклади сучасних розумних ліжок, розроблених для підтримки здорового сну людини, а також медичних ліжок.

Медичні функціональні ліжка для лежачих хворих – спеціальний вид меблів. Медичне ліжко незамінне для лежачого хворого, який переніс важку операцію, травму, перелом.

У реабілітаційний період пацієнт більшу частину часу проводить в горизонтальному положенні. Інваліду або людині, яка пережила важке захворювання, потрібні комфортні умови. Поставлену задачу повністю вирішує особливий медичний предмет інтер'єру – функціональні ліжка для лежачих хворих.

Зручні меблі розділена на окремі секції, кожна з яких дозволяє регулювати положення частин тіла. Механічний або електричний привід допомагає пацієнтові самостійно підняти або опустити голову, тазостегнову або ножну частину.

При користуванні функціональним ліжком, лежачий хворий може уникнути застійних явищ, нормалізувати роботу внутрішніх органів, стабілізувати кровообіг і поліпшити самопочуття. Вироби часто оснащуються додатковими комплектуючими, що підвищують зручність пацієнта. Це механізми для підтягування, бічні огорожі, тумби і столики. Наприклад, медичні ліжка дозволяють пацієнтові без зусиль прийняти позицію Фаулера, коли голова і плечі на 45-60° вище тіла. У такому положенні полегшується дихання, поліпшується робота шлунково-кишкового тракту, зменшується ймовірність виникнення застійних явищ.

Також на сьогодні на ринку вже існує декілька систем розумних ліжок, кожне з яких має свої переваги і свої недоліки. Розглянемо деякі з них.

Один із таких проектів – ліжко Balluga, яке створено Джо Катаном [3]. Основною його перевагою вважається матрац, який замість звичайних пружин наповнений системою сферичних комірок, які можуть здуватися чи надуватися окремо одна від одної, також у ліжка є клімат-контроль і функція вібромасажу, це все представлено на рисунку 1.1. Серед особливостей такого ліжка є:

- двоспальність;
- регулювання положення ліжка за рахунок фізичних навантажень на нього, що не завжди призводить до бажаного користувачем результату;

– керування зі смартфона.

Іншим аналогом є ліжко від Ендрю Росмана Sleep Number x12 [4], можливості якого представлені на рисунку 1.2.



Рисунок 1.1 – Особливості ліжка Balluga

Ліжко Sleep Number x12 надає перевагу більше зчитуванню стану користувачів, ніж механічним налаштуванням в залежності від положення людини, через що претендує на більшу, аніж попередній проект, спорідненість із інтернетом речей. Sleep Number x12 має такі датчики:

- датчик дихання;
- датчик хропіння;
- датчик серцевих скорочень користувача;
- датчик голосових команд;
- датчик тиску, який намагається визначити положення користувача під час сну;
- датчик руху.



Рисунок 1.2 – Особливості ліжка Sleep Number x12

Всі наведені датчики збирають інформацію, користуючись якою можна налагодити систему покращення сну користувача, але ліжко Sleep Number x12 не має впливу на фізичний стан користувача під час сну чи відпочинку в достатній мірі, воно корегує лише положення голови, якщо почує хрпіння, або, коли розпізнає необхідну команду, освітлення в кімнаті. Решта інформації іде на формування теоретичних порад щодо сну, які користувач стануть доступні користувачу лише після його відпочинку, і в більшості випадків повинен буде організовувати зміни самостійно.

### 1.3 Постановка задачі

На основі аналізу існуючих проблем, дійсних аналогів та вимог до них, сформулюємо мету роботи, як дослідження оптимізації стану користувача після сну та вибір найбільш такого варіанту оптимізації, що є найбільш придатним до предметної області, а також розробку власної моделі оптимізації відповідно до обраних методів.

В результаті атестаційної роботи має бути розроблена система розумного ліжка Beddest, яка за допомогою інструментів штучного інтелекту задовільнить потребам людини у здоровому і комфортному сні або відпочинку, і системи керування нею. Розглянемо детально інформаційний метаболізм програми, тобто потік інформації на вході і на виході з неї (див. рис. 1.3).

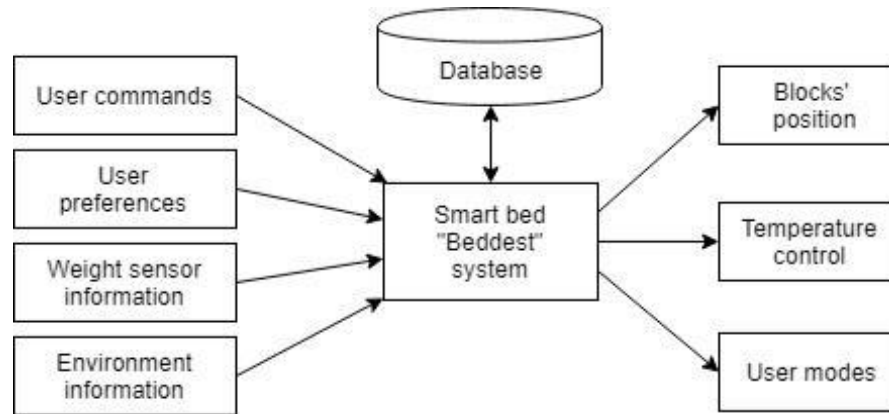


Рисунок 1.3 – Вхідна та вихідна інформація для роботи програми

На рисунку 1.3 представлено схему взаємодії системи із інформацією, що надходить і результуючою інформацією. В якості вхідних даних система Beddest приймає інформацію про стан користувача, його наявність та положення на ліжку, його вподобання, які були визначені в ході користування системою, та інформацію про навколишню середу. Логічна обробка інформації, що надійшла, і вже наявної інформації з бази даних дозволяють системі, яка розробляється, регулювати положення блоків матрацу, а отже і користувача на них, також контролювати температуру ліжка за бажанням користувача, а також зберігати нові режими ліжка, що найліпше підходять юзеру.

Щоб мінімізувати необхідність користувача брати участь в процесі вибору оптимального стану для сну, доцільно буде використовувати інструменти штучного інтелекту. Штучний інтелект системи, що розробляється, буде відповідати за оптимізації критеріїв впливу на сон користувача, важливою частиною планування роботи з системами штучного інтелекту є вибір оптимальної моделі надання знань [5]. Саме вибір оптимальної моделі дозволяють більш



ефективно працювати зі знаннями, що приходять і накопичуються, і на їх основі приходити до деякого рішення.

Іншими словами задачу роботи можна розбити на такі підзадачі:

- провести аналіз та моделювання проблемної області керування розумними ліжками;
- дослідити моделі надання знань;
- розробити математичну модель для керування розумним ліжком;
- створити базу даних для зберігання даних з розробленої моделі;
- спроектувати архітектуру системи;
- розробити алгоритми придбання знань та висновку для керування моделлю розумного ліжка;
- програмно реалізувати систему.

## 2 МАТЕМАТИЧНА МОДЕЛЬ ДЛЯ КЕРУВАННЯ РОЗУМНИМ ЛІЖКОМ

### 2.1 Аналіз та моделювання предметної області керування сном

Сон – це життєво необхідний, періодично наступаючий особливий функціональний стан організму, що займає у людини щодня близько 1/3 часу і характеризується відсутністю довільної рухової активності, майже повним відключенням від сенсорних впливів зовнішнього світу, сновидіннями. Безсумнівно, що сон служить для обробки інформації. У багатьох снах дорослих і дітей з'являються події і переживання, які відбулися незадовго до сну.

Сон поділяється на дві основні фази: парадоксальний сон, названий також сном зі швидкими рухами очей (або REM-сон), і сон без швидких рухів очей (або NREM-сон), які є послідовністю функціональних станів мозку – чотирьох стадій фази повільного сну. Стадії NREM-сну і REM-сну становлять один цикл сну, і таких циклів у здорової людини буває від чотирьох до шести за ніч. Ці цикли не однакові: в перших двох максимально представлені фази повільного сну, а в ранкових – фаза швидкого сну. REM-сон характеризує швидкими рухами очей, посиленою мозковою діяльністю і дуже низьким м'язовим тонусом. Рівень мозкової діяльності схожий з рівнем під час неспанья. У фазу швидкого сну можна спостерігати, як під закритими століттями рухаються очні яблука людини, також саме в цей час ми бачимо сновидіння. Часто людина неспокійний, дихання уривчасто, уста і пальці можуть смикатись, саме цей час найбільш підходить для пробудження. Такий сон настає кожні 60-90 хвилин, самі тривалі й інтенсивні періоди REM-сну доводиться приблизно на 5 ранку, коли температура тіла опускається до мінімуму. [6]

Сон також пов'язаний із так званими біологічними годинниками організму. До сну відносяться два основних ритму:

– циркадний ритм – цикл сну та неспанья протягом доби, є одним з найбільш виражених біологічних ритмів, визначає загальну активність і впливає на більшість фізіологічних систем організму людини [7];

– ультрадіадний ритм – цикл фаз повільного і швидкого сну. У дітей більша частина сну доводиться на швидкий сон, коли у дорослих він становить майже четверту частину всього сну. Цей ритм змінюється протягом ночі і триває в середньому близько півтори години у дорослих [7];

Тривалість сну індивідуальна у кожної людини. Вона може залежати від темпераменту, психофізіологічного стану та ряду інших причин, тобто в силу індивідуальних особливостей одним людям потрібна менша кількість сну, ніж іншим.

Відомо, що депривація сну призводить до порушень у багатьох сферах життя і здоров'я, а повне припинення сну веде до загибелі експериментальних тварин. На сьогоднішній день не можна з остаточною впевненістю сказати, навіщо людина спить, але можна сміливо стверджувати, що будь-які порушення в циклі «сон-неспання» тягнуть за собою порушення метаболічних і обмінних процесів, що, в свою чергу, значно впливає на якість життя людини. [8]

Є багато положень, в яких людина може спати, власне, їх кількість обмежується лише гнучкістю конкретної людини і його фантазією, однак далеко не всі вони можуть дозволити людині добре виспатися, і, головне, не шкодувати весь день після цього, що вибрав саме це положення.

Розглянемо найбільш загальні з них:

Сон на спині. Незважаючи на вкрай високу популярність на інтернет-форумах, такий стан має свої недоліки, а не тільки переваги.

Переваги:

– завдяки правильному положенню голови щодо тіла зменшує ризик виникнення печії;

– підтримує нейтральне положення хребта, що сприяє найбільшому розслабленню під час сну;

– дозволяє уникнути зайвого впливу на шкіру і м'язи обличчя, що сприяє більш свіжому зовнішньому вигляду після пробудження.

Недоліки:

– неправильне положення нижньої щелепи може призвести до утруднення дихання, через це такий стан зазвичай протипоказаний людям із захворюваннями дихальних шляхів;

– для вагітних жінок сон на спині може відгукнутися болем в попереку на наступний день, це також пов'язано з положенням хребта при додатковому навантаженні.

Сон на боці. Наступне найбільш популярне положення під час сну.

Переваги:

– знижує ризик появи печії за тим же принципом, що був приведений в списку плюсів положення на спині;

– забезпечує оптимальне положення шиї і спини за умови, що була правильно підібрана подушка;

– сон на лівому боці покращує кровообіг в організмі, що особливо корисно для вагітних;

– зменшує можливість появи хропіння;

– знімає м'язову напругу в районі попереку у вагітних жінок.

Недоліки:

– несприятливий вплив зовнішнього середовища на шкіру і м'язи обличчя: обличчя приймає неприродне положення при зіткненні з подушкою, що може викликати кілька «пом'ятій» вигляд після пробудження.

Важливо також зауважити, що положення хребта залежить в рівній мірі і від твердості матрацу: занадто м'який матрац буде нерівномірно вигинатися під центром ваги тіла людини, в результаті чого вона буде спати у вигнутому положенні, занадто твердий же матрац буде змушувати стегно і плече сильно підніматися над талією, що також призведе до вигинання хребта.

Сон на животі. Більшість фахівців погоджуються в тому, що ця поза є найменш сприятливою для здорового сну. Коли людина приймає таке положення, хребет вигинається під неправильним кутом, поверхня обличчя повністю контактує подушкою, що сприяє поганому самопочуттю та вигляду вранці.

#### Переваги:

– досить ефективно протистоїть виникненню хропіння, знижуючи таку можливість до мінімуму через оптимальне розташування дихальних шляхів в такому положенні.

#### Недоліки:

– неможливість підтримувати природні прогини хребта, тобто його оптимальне положення, що може призвести в деяких випадках навіть до розтягування, защемлення нервів та ін.;

– контакт шкіри обличчя, шиї і грудей із ліжком, тобто зовнішнім середовищем максимальний, це призводить до тих же результатів, що описані в переліку недоліків положення на боці.

Потрібно зауважити, що теорія правильного положення під час сну базується не тільки на положенні тіла на матраці, але включає в себе й розташування і якість подушки або системи подушок в ліжку.

Наприклад, якщо навіть під час сну на спині людини переслідують больові відчуття в ногах або в районі попереку, то можна використовувати додатковий реквізит – невеликі подушки, валики або згорнуті в рулон рушники відмінно підходять для розміщення під коліна або під поперековий відділ хребта для зняття напруги в відповідних зонах.

При болях у плечі також радять прийняти положення на боці, на здоровій стороні і притиснути подушку до грудей обома руками. Таке положення зможе забезпечити ушкодженому місцю найбільш зручний стан, що, в свою чергу, може зменшити біль.

Болі в тазостегнових суглобах може викликати ситуація, коли, прийнявши положення на боці, людина відсуває вільну (умовно кажучи, верхню) ногу трохи вперед, викликаючи обертання попереково-крижового відділу хребта. Однак і цю проблему легко вирішити, затиснувши невелику подушку або рулон між колінами.

Підкладання подушки або подушок під ноги так, щоб серйозно збільшити різницю в висоті ніг щодо решти тіла, також показано людям із захворюваннями,

пов'язаними з підвищеним тиском в кінцівках або для більш ефективного зняття втоми ніг.

Також важливою частиною цієї теорії є вибір правильної подушки. Подушка, як традиційний постільний предмет, спеціальний рулон, іграшка або згорнутий рушник, яку підкладають під голову, призначена для підтримки нейтрального і, як наслідок, найбільш комфортного положення шийного відділу хребта і голови без будь-яких перекосів. [9]

В ідеалі для кожного з описаних вище положень і подушка повинна відповідати відповідним критеріям, що будуть розглянуті нижче, але мало хто вважатиме зручним мати кілька подушок і змінювати їх одну на іншу під час сну (або, якщо більш реалістично, відпочинку) по мірі зміни положення. Фахівці рекомендують запам'ятовувати, в яких положеннях людина найчастіше прокидається – це і будуть найбільш зручні положення для сну, і подушку рекомендують обирати саме для цього положення, що, зрозуміло, не скасовує того, що така подушка може бути абсолютно незручною для іншого положення, яке може бути прийнято уві сні.

Отже, для підтримки правильного розташування хребта під час лежання на спині, необхідно, щоб голова лежала прямо, а шия не згиналася ні вперед, ні назад, тобто подушка для цього положення не повинна бути занадто низькою або занадто високою, в ідеалі – повинна мати спеціальне поглиблення в районі потилиці.

Відпочинок в положенні на боці найкращим чином впливає на організм людини за умови, що хребет, шия і основу черепа будуть утворювати пряму лінію, паралельну поверхні ліжка. Для цього подушка повинна займати рівно той простір, що залишається між матрацом і головою, причому необхідно враховувати і товщину руки, якщо людина уві сні підкладає її під подушку, і вибирати ще більш тонку.

Якщо людині все ж зручніше спати на животі, то рекомендується вибрати або дуже тонку подушку, або відмовитися від неї зовсім, щоб звести негативні наслідки від викривлення хребта до мінімуму.

Додатково найвдаліші положення варто було б запам'ятовувати, щоб використовувати їх і вподальшому. Також корисним буде розуміння, які положення вважаються найпопулярнішими серед людей приблизно такої ж віку та комплекції.

Підбиваючи підсумок, наведемо Use-case діаграму для наочного представлення функціоналу, який може вирішувати система, що розробляється, при використанні підходів, що були розглянуті вище (див. рис. 2.1).

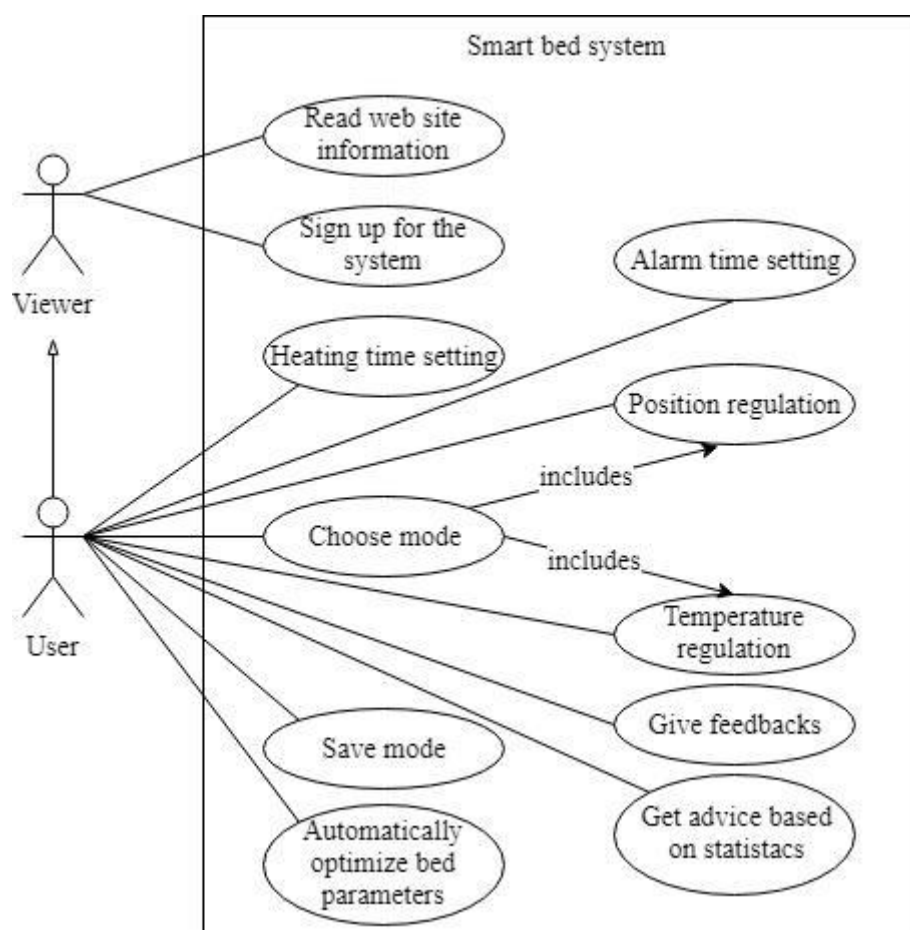


Рисунок 2.1 – Діаграма прецедентів

Згідно із діаграмою будь-яка людина може продивитися інформацію про продукт або зареєструватися в системі, коли можливості користувач набагато більші: він може як задавати параметри ліжка вручну, так і передавати керування станом ліжка системі, запам'ятовувати улюблені положення та застосовувати їх, а також отримувати поради щодо стану ліжок, що засновані на статистичних висновках.

## 2.2 Розробка моделі розумного ліжка

Припустимо, що весь матрац, який стане частиною розумного ліжка, складається з окремих блоків фіксованого розміру, які кріпляться до основи ліжка. В цьому випадку, якщо закріпити на кожному з таких блоків механізм підняття і повороту блоку уздовж довжини ліжка, і розмістити ці блоки на незначному відстані один від одного для більшої рухливості, можемо отримати ліжко, яке в змозі задовольняти деяким вимогам методів впливу на сон: регулювання положення частин матраца (див. рис. 2.2).



Рисунок 2.2 – Схема блоку з регульованим положенням

Регулятор висоти являє собою стрижень, що збільшується за принципом «матрьошки», нижня частина якого кріпиться до основи ліжка, верхня – до регулятора кута нахилу. Регулятор кута нахилу – механізм, під дією якого горизонтальна пластина, що кріпиться верхньою частиною до самого блоку матраца, змінює кут нахилу щодо початкового стану.

Теорії про положення тіла також говорять про те, що в різних випадках необхідна різна твердість ліжка, іншими словами, тепер необхідно зробити блоки з регульованим положенням блоками з регульованою щільністю, що для користувача буде позначати регулювання твердості блоку (див. рис. 2.3, 2.4).



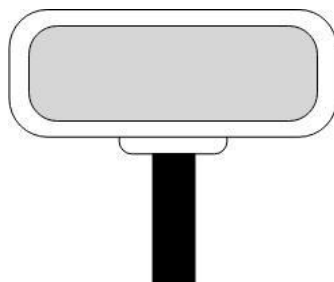


Рисунок 2.3 – Схема м'якого блоку, що регулюються

На рисунку 2.3 зображена схема блоку з регулюванням положенням і регулюванням твердості. Внутрішній округлений прямокутник позначає порожнє тіло з міцного еластичного матеріалу. Саме в нього закачуватиметься матеріал, який регулює щільність внутрішності, а отже, і твердість блоку. Матеріал повинен надавати рівномірний тиск на стінки еластичного тіла, маса матеріалу в обмеженому обсязі повинна бути обмеженою так, щоб значення тиску не наближатися до граничного. В якості такого матеріалу можна використовувати повітря, на малюнку він позначений сірою областю. Проблема полягає в тому, щоб надати внутрішньому тілу форму блоку і захистити від зовнішніх впливів, з цим справляється зовнішня частина блоку. Зовнішній округлений прямокутник – схематичне позначення саме такої зовнішньої частини, яка утримує внутрішнє тіло в необхідній формі, а також не дає силам зовнішнього середовища діяти на нього.

На рисунку 2.4 зображена схема такого ж блоку, що і на рисунку 2.2, з однією відмінністю: додатковими шарами зовнішнього захисту. Такі блоки будуть витримувати більший тиск, а, отже, і твердість забезпечувати більше, ніж м'який регульований блок.

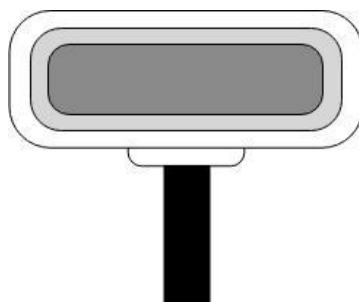


Рисунок 2.4 – Схема жорсткого блоку, що регулюються

Блоки розташовані один поряд із одним на нерухомій основі, як показано на рисунку 2.5. В одній комплектації ліжка можуть знаходитися як більш м'які, так і жорсткіші блоки.

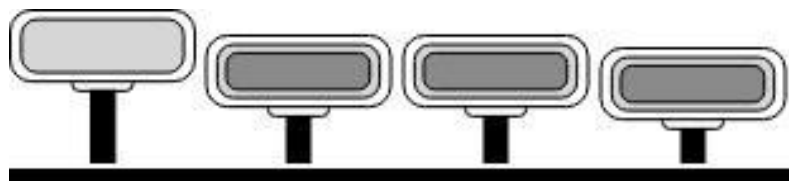


Рисунок 2.5 – Схема ліжка із блоками матрацу

Таким чином будь-який блок ліжка може регулюватися окремо від інших за бажанням користувача.

Сам предмет інтернету речей буде складатися з декількох запчастин, для нормального функціонування необхідно буде купити їх всі. Тобто для придбання ліжка потрібно закупити:

- основу для самого ліжка;
- набір регульованих блоків необхідної твердості;
- ковдру з нагрівальними елементами.

### 2.3 Дослідження моделей надання знань

Інтелект (від лат. *Intellectus* – відчуття, сприйняття, розуміння, поняття, розум), або розум – властивість психіки, що складається із здатності пристосовуватися до нових ситуацій, здатності до навчання і запам'ятовування на основі досвіду, розуміння і застосування абстрактних концепцій і використанню своїх знань для управління навколишнім середовищем. Інтелект – це загальна здатність до пізнання і вирішення труднощів, яка об'єднує всі пізнавальні здібності людини: відчуття, сприйняття, пам'ять, мислення, уява.

Штучний інтелект – це область інформаційних технологій, яка займається розробкою інтелектуальних комп'ютерних систем, тобто систем, що володіють можливостями, які ми традиційно пов'язуємо з людським розумом, – розуміння мови, навчання, здатність міркувати, вирішувати проблеми та ін. Пізніше до ШІ стали відносити ряд алгоритмів і програмних систем, відмітною властивістю яких є можливість вирішувати різноманітні завдання так, як це робить людина, що розмірковує над їх вирішенням.

У штучному інтелекті основна мета – навчитися зберігати знання таким чином, щоб програми могли обробляти їх і досягти подоби людського інтелекту. Дослідники ШІ використовують теорії подання знань з когнітології, тобто використовуються теорії обробки інформації людиною, щоб отримати такі методи як фрейми, правила і семантичні мережі прийшли в ШІ з теорій обробки інформації людиною [10]. Так як знання використовується для досягнення розумної поведінки, фундаментальною метою дисципліни надання знань є пошук таких способів надання, які роблять можливим процес логічного висновку, тобто створення висновків з знань.

Моделі подання або надання знань – одне з найважливіших напрямків досліджень в області штучного інтелекту. Без знань штучний інтелект не може існувати, а саме моделі надання задають структуру зберігання даних, які згодом будуть оброблятися для прийняття штучним інтелектом рішення. На сьогоднішній день розроблено вже достатня кількість моделей, кожна з них має свої плюси і мінусами, і тому для кожного конкретного завдання необхідно вибрати саме свою модель. Від цього буде залежати не тільки ефективність виконання поставленого завдання, але і можливість її вирішення взагалі.

На рисунку 2.6 наведена класифікація моделей надання знань.

Види моделей надання знань можна розділити на теоретичні та емпіричні.

Перший підхід можна визначити як теоретично обґрунтований, тобто той, що гарантує правильність рішень. Він в основному представлений моделями, заснованими на формальній логіці (обчислення висловлювань, числення

предикатів), формальних граматиках (спосіб опису формальної мови, тобто виділення деякої підмножини з безлічі всіх слів деякого кінцевого алфавіту), комбінаторних моделях, зокрема моделях кінцевих проектних геометрій, теорії графів, алгебраїчних моделях та ін.

В логічній моделі вся інформація розглядається як сукупність фактів і тверджень, що їх зв'язують, які представляються як формули в деякій логіці. Знання при цьому надаються набором подібних одне до одного тверджень, а побудова висновків і отримання нових знань зводиться до реалізації процедури логічного висновку. Цей процес може бути строго формалізований, тому що в його основі лежить класичний апарат математичної логіки.

Комбінаторні моделі засновані на аналізі дискретних об'єктів, кінцевих множин і заданому на них відношенню порядку. В рамках комбінаторики також аналізуються всі можливі зміни, перестановки і поєднання, в рамках заданих множин. Под комбінаторикою розуміють більш великий розділ дискретної математики, що включає, зокрема, теорію графів. Комбінаторні моделі використовуються в задачах топології (наприклад, пошук шляху), завданнях прогнозування поведінки автоматів, при вивченні дерев рішень, частково впорядкованих множин.

Алгебраїчна модель передбачає надання знань у вигляді деяких алгебраїчних примітивів, над якими визначено безліч дій (деякі з яких можна задати таблично). Для набору знань, що представлений в такому вигляді, діють правила алгебраїчних множин, такі як формалізація, визначення підсистем і відносин еквівалентності. Також можлива побудова ланцюгів множин (безлічі, для яких визначено порядок відносини «бути підсистемою»).

В рамках теоретичного підходу до теперішнього часу вдавалося вирішувати тільки порівняно прості завдання з вузькою предметною областю.

Інший підхід, що зветься емпіричним, заснований на вивченні принципів організації людської пам'яті і моделюванні механізмів вирішення завдань

людиною. На основі цього підходу в даний час розроблено велику кількість видів моделей, деякі з них ми розглянемо далі.

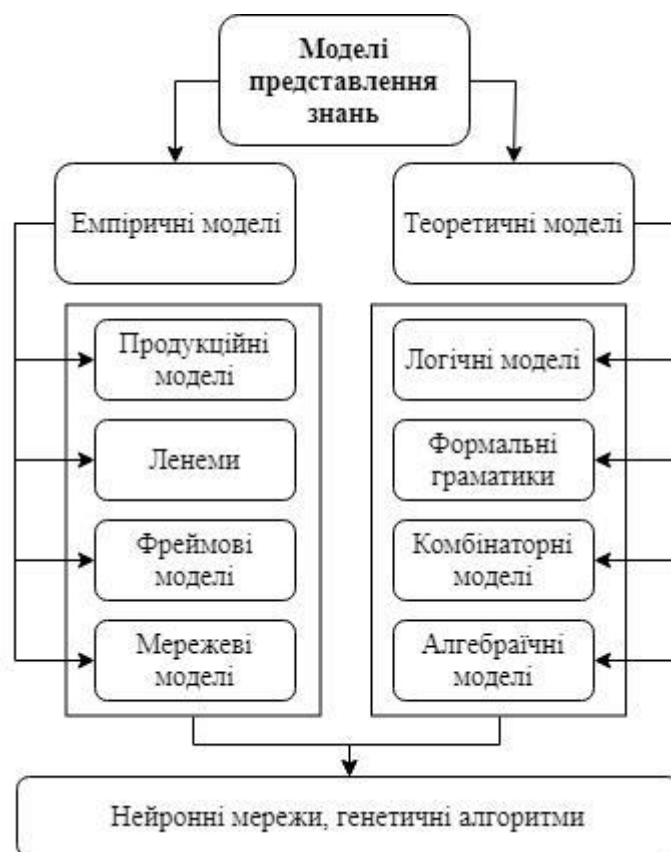


Рисунок 2.6 – Класифікація моделей надання знань в ШІ

Інший підхід, що зветься емпіричним, заснований на вивченні принципів організації людської пам'яті і моделюванні механізмів вирішення завдань людиною. На основі цього підходу в даний час розроблено велику кількість видів моделей, деякі з них ми розглянемо далі.

Мережеві моделі (або семантичні мережі) – інформаційна модель предметної області, що має вигляд орієнтованого графа, вершини якого відповідають об'єктам предметної області, а дуги (ребра) задають відносини між ними. Розрізняють екстенціональні й інтенціональні семантичні мережі. Екстенціональна семантична мережа описує конкретні зв'язки даної ситуації. Іntenціональна – імена класів об'єктів, а не індивідуальні імена об'єктів. Зв'язки в інтенціональних мережах відображають ті відносини, які завжди притаманні об'єктам даного класу.

Продукційна модель – це уявлення знань, засноване на правилах, побудованих на використанні виразів виду «ЯКЩО (умова), ТО (дія)». В даний час є найбільш проробленою і поширеною моделлю представлення знань, особливо в експертних системах.

Фреймова модель ґрунтується на такому понятті як фрейм. Фрейм – структура даних для представлення деякого концептуального об'єкта. Інформація, що відноситься до фрейму, міститься в слотах, з яких він складається. Фрейми можуть поділятися на фрейм-екземпляр (конкретна реалізація фрейму, що описує поточний стан в предметній області), фрейм-зразок (шаблон для опису об'єктів або допустимих ситуацій предметної області), фрейм-клас (фрейм верхнього рівня для представлення сукупності фреймів зразків).

Ленеми являють собою змішаний тип моделі, що логічним продовженням, розвитком інших емпіричних моделей. Ленема призначена для структурного комплексного опису понять предметної області. За образотворчим можливостям ленеми більш досконалі, ніж такі традиційні моделі подання знань, як семантична мережа, фрейм або система продукцій. Однак, для деяких понять подання знань на основі ленеми може бути незручним і навіть неприйнятним. Наприклад, такі поняття, в описі яких дуже велику роль грає внутрішня динаміка.

Ідея ж генетичних алгоритмів запозичена у живої природи і складається в організації еволюційного процесу, кінцевою метою якого є отримання оптимального рішення в складній комбінаторній задачі. Розробник генетичних алгоритмів виступає в даному випадку як "творець", який повинен правильно встановити закони еволюції, щоб досягти бажаної мети якомога швидше.

На вхід генетичні алгоритми можуть отримувати будь-яку модель, і самі по собі є скоріше не моделлю подання знань, а моделлю роботи з ними, їх обробки подальшої обробки.

Зробимо аналіз наведених моделей подання знань за критеріями, що необхідні для максимально ефективною реалізації знаходження оптимального стану сну для системи розумного ліжка:

- здатність обробляти окремі параметри;
- здатність досить чітко для швидкого прийняття рішення визначити залежність між вхідними параметрами та очікуваним результатом;
- якомога простіша підтримка розвитку, набуття знань в режимі реального часу;
- мінімальні потреби для зберігання логіки.

Результати аналізу наведені у таблиці 2.1.

Таблиця 2.1 – Порівняння моделей подання знань

Моделі знань	Окремі параметри	Чітка залежність результату	Підтримка навчання	Простота логічного виводу	Мінімальні потреби зберігання
Логічні моделі	+	+	–	–	+
Формальні граматики	+	+	–	–	–
Комбінаторні моделі	–	–	–	–	–
Алгебраїчні моделі	–	–	–	–	–
Продукційні моделі	+	+	+	+	+
Ленеми	+	–	–	–	–
Фреймові моделі	+	–	+	+	+
Мережеві моделі	–	–	+	+	+

Виходячи із наведеної таблиці можна прийти до висновку, що найбільш доцільним для розроблюваної системи буде використання продукційних моделей подання знань. Саме ця модель відповідає усім основним виявленим критеріям.

## 2.4 Дослідження продукційної моделі подання знань

У загальному вигляді під продукцією розуміється вираз такого вигляду:

(i); Q; p:  $A \Rightarrow B$ ; N.

Тут і – ім'я продукції, за допомогою якого дана продукція виділяється зі всієї безлічі продукцій. Елемент Q характеризує сферу застосування продукції. Основним елементом продукції є її ядро:  $A \Rightarrow B$ . Інтерпретація ядра продукції може бути різною і залежить від того, що стоїть ліворуч і праворуч від знаку секвенції (послідовності)  $\Rightarrow$ . Звичайною прочитання ядра продукції виглядає так: ЯКЩО А, ТО В, більш складні конструкції ядра допускають в правій частині альтернативний вибір, наприклад, ЯКЩО А, ТО В1, ІНАКШЕ В2. Іноді результати В1 та В2 подаються із коефіцієнтом ефективності для додання чіткості системі. Секвенція може тлумачитися в звичайному логічному сенсі як знак логічного слідування В з істинного А (якщо А не є істинним виразом, то про В нічого сказати не можна). Можливі й інші інтерпретації ядра продукції, наприклад, А описує деяку умову, необхідну для того, щоб можна було вчинити дію В.

Елемент Р є умова застосовності ядра продукції. Зазвичай Р являє собою логічне вираження (як правило, предикат). Коли Р приймає значення "істина", ядро продукції активізується. Якщо Р помилково, то ядро продукції не може бути використано.

Елемент N описує постумови продукції. Вони актуалізуються тільки в тому випадку, якщо ядро продукції реалізувалося.

Якщо в пам'яті системи зберігається певний набір продукцій, то вони утворюють систему продукцій. В системі продукцій повинні бути задані спеціальні процедури управління продукціями, за допомогою яких відбувається актуалізація продукцій і вибір для виконання тієї чи іншої продукції з числа актуалізованих.

Існують два типи продукційних систем – з «прямими» і «зворотніми» висновками. Прямі висновки реалізують стратегію «від фактів до висновків». При



зворотних висновках висуваються гіпотези ймовірнісних висновків, які можуть бути підтверджені або спростовані на підставі фактів, що надходять в робочу пам'ять. Існують також системи з двонаправленими висновками.

У прямих висновках обирається один з елементів даних, що містяться в базі даних, і якщо при порівнянні цей елемент узгоджується з лівою частиною правила (посилкою), то з правила виводиться відповідний висновок і поміщається в базу даних або виконується дія, яка визначається правилом, і відповідним чином змінюється вміст бази даних.

У зворотних висновках процес починається від поставленої мети. Якщо ця мета узгоджується з правою частиною правила (висновком), то посилка правила приймається за підциль або гіпотезу. Цей процес повторюється до тих пір, доки не буде отримано збіг підцилі з даними.

Розглянемо роботу прямих та зворотніх висновків на прикладі. Уявімо, що в нас є декілька правил про вибір комп'ютеру:

– ЯКЩО { потрібна мобільність } ТА { сучасне обладнання }, ТО { необхідний ноутбук }, ПІСЛЯ ЦЬОГО { замовити ноутбук };

– ЯКЩО { працюєш в різних місцях }, ТО { потрібна мобільність }.

Припустимо, що на вхід системи надійшли такі дані: користувач потребує сучасне обладнання та працює не на одному місці.

Проілюструємо порядок виконання прямого висновку на наданому прикладі.

Крок 1. Розпочинаємо перший цикл проходів:

– пробуємо П1, не можемо бути певні через нестачу даних – не можемо сказати, чи потрібна мобільність;

– пробуємо П2, бачимо, що правило працює, тобто до даних додається “потрібна мобільність”.

Крок 2. Розпочинаємо другий цикл проходів:

– пробуємо П1, бачимо, що правило працює, результат знайдений – “потрібен ноутбук”. Після цього може також виконитися постумова “замовити ноутбук”.

Проілюструємо порядок виконання зворотнього висновку на тому ж самому прикладі.

В цьому випадку на вхід у систему подається ціль користувача: купити ноутбук, також ми знаємо, що користувач працює у різних місцях:

- розпочинаємо перший цикл проходів;
- знаходимо П1 із ціллю “купити ноутбук”, не можемо бути певні через нестачу даних – не можемо сказати, чи потрібна мобільність. Ставимо за нову ціль отримання даних про мобільність;
- знаходимо П2 із ціллю “потрібна мобільність”, бачимо, що умови для правила збігаються зі вхідними умовами, тобто правило працює. Тож ціль “потрібна мобільність” активована;
- розпочинаємо другий цикл проходів;
- повертаємося до П1, бачимо, що правило працює, результат “потрібен ноутбук” активовано. Після цього може також виконатися постумова “замовити ноутбук”.

Також розглянемо варіанти логічного висновку з продукційних моделей.

Пошук в глибину. При пошуку рішення як чергова підціль вибирається та, яка відповідає наступному, більш детальному рівню опису завдання. Наприклад, діагностична система, зробивши на основі відомих симптомів припущення про наявність певного захворювання, буде продовжувати запитувати додаткові ознаки і симптоми цієї хвороби до тих пір, доки повністю не доведе або спростує висунуту гіпотезу:

Пошук в ширину. При пошуку в ширину, навпаки, система спочатку аналізує умови, що також називаються класифікаційними ознаками, одного рівня деталізації, а потім переходить до наступного (більш детального) рівня. Приведемо приклад: система, що діагностує хвороби, спочатку має проаналізувати всі симптоми, що знаходяться на першому рівні деталізації, навіть якщо вони належать до різних захворювань, зробить висновки, і тільки після цього перейде до симптомів наступного рівня.

Принцип найбільш довгої умови (різновид пошуку в глибину). Полягає у виборі з набору готових продукцій тієї, у якій стало справжнім найбільш «довге» умова здійсненості ядра. Цей принцип спирається на міркування «здорового глузду», що приватні правила, що відносяться до вузького класу ситуацій, важливіше загальних правил, що відносяться до широкого класу ситуацій, так як перші враховують більше інформації про ситуацію, ніж інші.

Принцип метапродукцій. Заснований на ідеї введення в систему продукцій спеціальних метапродукцій, завданням яких є організація управління при можливості неоднозначного вибору з набору готових продукцій. Якщо простіше, то при отриманні деяких вхідних умов, деякі висновки стають більш пріоритетними, ніж інші.

Наведемо сильні і слабкі сторони систем продукцій.

Продукції в порівнянні з іншими формами надання знань мають наступні переваги:

- модульність;
- однаковість структури (основні компоненти продукційної моделі можуть застосовуватися для побудови інтелектуальних систем з різною проблемною орієнтацією);
- природність (висновок укладення в продукційної моделі багато в чому аналогічний процесу міркувань експерта);
- гнучкість родовидної ієрархії понять, яка підтримується тільки як зв'язок між правилами (зміна правила веде за собою зміну в ієрархії);
- простота створення і розуміння окремих правил;
- простота поповнення і модифікації;
- простота механізму логічного висновку.

Слабкі сторони систем продукцій:

- процес виведення менш ефективний, ніж в інших системах, оскільки велика частина часу при виведенні витрачається на непродуктивну перевірку застосовності правил;

- складно уявити родовидні ієрархію понять;
- неясність взаємних відносин правил;
- складність оцінки цілісного образу знань;
- відміну від людської структури знань;
- відсутність гнучкості в логічному висновку.

## 2.5 Розробка моделі для керування розумним ліжком

Для створення штучного інтелекту, який буде займатися вибором оптимального положення розумного ліжка для сну, з використанням продукційних моделей в якості моделі надання знань, необхідно скласти список характеристик, які:

- знаходяться під впливом системи;
- є незалежними параметрами, як, наприклад, характеристики середовища або користувача;
- цільові параметри сну, на які передбачається вплив.

В рамках дослідження була розроблена модель для керування розумним ліжком.

Так оптимальний стан розумного ліжка  $S^{opt}$  для забезпечення сну необхідної якості можна задати на базі наступної формули [11]:

$$S^{opt} = \text{extremum } Q(S), \quad (1)$$

де  $Q$  – принцип оптимізації, який можна задати за допомогою продукційної моделі;

$S$  – сон та його параметри;

$S^{opt}$  – оптимальний стан розумного ліжка.

Представимо модель сну людини як:

$$S = \langle B, P, W, TC \rangle, \quad (2)$$

де  $B$  – стан ліжка;

$P$  – стан людини;

$W$  – погодні умови (умови навколишнього середовища);

$TC$  – часові залежності.

При цьому стан людини-користувача включає ряд параметрів:

$$= \{ , , , h, , \}, \quad (3)$$

де  $A$  – вік користувача;

$G$  – стать користувача;

$w$  – вага користувача;

$h$  – зріст користувача;

$RP$  – один із результуючих параметрів, який користувач завдав як пріоритетний;

$pref$  – власні вподобання людини щодо стану ліжка.

$$= \{ пр, зов, , , зов-мин, мин, зов-майб, майб \}, \quad (4)$$

де  $пр$  – температура у приміщенні;  
 $зов$  – зовнішня температура;

– атмосферний тиск;

– опади;

$зов-мин$  – різниця температур від учора на сьогодні;  
 $мин$  – різниця опадів від учора на сьогодні;  
 $зов-майб$  – різниця температур від сьогодні на завтра;  
 $майб$  – різниця опадів від сьогодні на завтра.

Усі параметри, що відносяться до погодних умов, будуть отримуватися з мережі, та в системі будуть розглядатися як незмінювані дані.

$$= \{ \text{зас проб, проб, сну, сну-мин, несп-мин} \}, \quad (5)$$

де зас – час засипання;

проб – час пробудження;

проб – різниця між часом дзвінка будильника та часом пробудження;

сну – тривалість сну;

сну-мин – кількість сну в попередній період;

несп-мин – кількість годин неспання в попередній період.

Усі ці параметри, окрім тривалості сну, будуть враховуватися при знаходженні оптимального положення юзеру [12]. Тривалість сну буде оброблятися, коли користувач буде давати свій відзив про якість сну.

В рамках дослідження розроблена модель ліжка, яка буде використана як основа для системи розумного ліжка. Увесь матрац, який стане частиною розумного ліжка, складається з окремих блоків фіксованого розміру, які кріпляться до основи ліжка. На кожному з таких блоків закріплений механізм підняття і повороту блоку вздовж довжини ліжка, вони розміщені на незначній відстані один від одного для більшої рухливості [13]. Отже, стан ліжка визначається наступним чином:

$$= \{ p, c, n, v_n \}, \quad (6)$$

де  $V_p$  – стан подушки;

$V_c$  – стан середніх блоків;

$V_n$  – стан блоку матрацу для ніг;

$T_{v_n}$  – температура ліжка.

Параметри блоку-подушки та блоку для ніг включають в себе інформацію про висоту, кут нахилу та м'якість блоку, середніх блоків – лише висоту та м'якість:

$$p = \{p, p, p\},$$

$$c = \{c, c\},$$

$$n = \{n, n, n\},$$

(7)

де  $p$  – параметри блоку-подушки;  
 $c$  – параметри середніх блоків;  
 $n$  – параметри блоку для ніг.

Параметри блоку-подушки та блоку для ніг включають в себе інформацію про висоту ( ), кут нахилу ( ) та м'якість блоку ( ). Параметри середніх блоків включають в себе тільки інформацію про висоту та м'якість. Внесемо логічні обмеження на ці параметри: висота не має перевищувати 20 см, кут нахилу в обидві сторони відносно довжини ліжка не має перевищувати 30°. Розташування атрибутів постілі також має залежати від положення тіла користувача.

Тож можна визначити, що якість сну залежить від параметрів його моделі, тобто параметрів погоди, людини-користувача, налаштувань самого розумного ліжка, а також часових параметрів. Тепер розглянемо, як саме можна оцінити якість сну. Позначимо результуючу якість сну літерою  $R$  та визначимо формулу залежності для неї:

$$R = \{ , , , , \}, \quad (8)$$

де (strength of dreaming) – глибина сну;

(drowsiness) – сонливість;

(fatigue) – втома;

(muscle pain) – біль у м'язах;

(headache) – головний біль.

Оптимальним станом є максимізована міцність сну та мінімізовані всі інші результуючі параметри якості сну.

Зведемо усі розглянуті параметри до однієї таблиці (див. табл. 2.2).

Таблиця 2.2 – Характеристики системи

Вхідні параметри		
параметри середовища	параметри сну	налаштування ліжка
температура приміщенні ( $T_{пр}$ ); температура на вулиці ( $T_{зов}$ ); атмосферної тиск ( $p$ ); опади ( $P$ ); різниця показників погоди зі вчора на сьогодні ( $\Delta T_{зов-мин}$ , $P_{мин}$ ); різниця показників погоди з сьогодні на завтра ( $\Delta T_{зов-майб}$ , $P_{майб}$ ).	в час засипання ( $t_{зас}$ ); час пробудження ( $t_{проб}$ ); середня різниця в часі між дзвінком будильника і пробудженням ( $\Delta t_{проб}$ ); міцність сну ( $SD$ ); тривалість сну ( $\Delta t_{сну}$ ); кількість сну в попередній період ( $\Delta t_{сну-мин}$ ); час з попереднього періоду сну ( $\Delta t_{несп--мин}$ ).	параметри блоку-подушки (висота, кут нахилу, м'якість) ( $H_{п}$ , $A_{п}$ , $S_{п}$ ); параметри середніх блоків (висота, м'якість) ( $H_{с}$ , $S_{с}$ ); параметри блоку для ніг (висота, кут нахилу, м'якість) ( $H_{н}$ , $A_{н}$ , $S_{н}$ ); температура між матрацом і ковдрою ( $T_{вн}$ ); час будильника ( $t_{буд}$ ).

Також розглянемо докладніше наведені характеристики і їх вплив на сон людини.

Температура в приміщенні ( $T_{пр}$ ) для сну впливає на якість сну через простоту дихання: оптимальної рівнем вважається температура нижча за температуру тіла людини на 10-15 градусів. Наприклад, в разі, якщо температура істотно вище або



нижче оптимального рівня, у користувача спостерігаються перебої з підтриманням стану спокою, що значно впливає на рівень бадьорості після пробудження.

Погодні умови, як атмосферний тиск ( $p$ ), опади ( $P$ ), зовнішня температура під час сну ( $T_{\text{зов}}$ ), а також температурна різниця між періодом сну і періодами 12 годин до ( $\Delta T_{\text{зов-мин}}$ ,  $P_{\text{мин}}$ ) та після періоду сну ( $\Delta T_{\text{зов-майб}}$ ,  $P_{\text{майб}}$ ) впливає на загальний стан користувача. Тут особливо важливо розглядати індивідуальну реакцію користувача на певні погодні умови і враховувати ці дані в подальшому при складанні оптимального періоду сну і стану ліжка під час нього.

До параметрів сну відносяться залежні і незалежні змінні, без урахування яких оцінювати якість сну неможливо. Основними параметрами є час засинання ( $t_{\text{зас}}$ ) і пробудження від сну користувача ( $t_{\text{проб}}$ ). Хоча реакція організму на сон в певні години доби можуть мати індивідуальні відмінності, існують загальні правила організації часу сну.

Різниця між часом дзвінка будильника і часом пробудження ( $\Delta t_{\text{проб}}$ ) може бути додатковою характеристикою оцінки загальної якості сну: чим краще користувач виспався за період відпочинку, тим швидше він буде прокидатися після дзвінка будильника.

Під міцністю сну ( $SD$ ) мається на увазі спокій користувача в період між засинанням і пробудженням, оцінюється з точки зору частоти зміни положення тіла. Міцність сну є незалежною змінною, показником спокою користувача в період відпочинку.

Різниця часу між засинанням і пробудженням ( $\Delta t_{\text{сну}}$ ) дозволяє дати оцінку безпосередньо тривалості відпочинку при інших умовах. Наприклад, при неправильному положенні шиї після пробудження самопочуття може бути тим гірше, чим більше була тривалість сну, в той же час при оптимальному стані в період сну кількість сну прямо пропорційно самопочуттю користувача після пробудження.

Кількість сну в попередній період ( $\Delta t_{\text{сну-мин}}$ ), як і час, що минув з попереднього відпочинку ( $\Delta t_{\text{сну-мин}}$ ), є показником рівня необхідності сну – депривації або

пересичення, і може допомагає у виборі оптимального часового проміжку для сну і підборі відповідного стану системи.

Попередньо була розроблена система розумного ліжка Beddest, яка дозволяє керувати параметрами ліжка, які описані в таблиці 2.1.

Після пробудження користувач може оцінити якість свого сну для удосконалення правил продукційних моделей підбору оптимального стану системи [14]. Оцінка відбувається в декількох розрізах:

- чи виспався користувач, або бажання продовжити сон присутнє (D – drowsiness);
- чи відчуває він бадьорість або втому (F – fatigue);
- чи відчуває користувач біль в м'язах шиї, спини, плечей, рук (MP – muscular pain);
- чи відчуває користувач головний біль, або запаморочення (HA – headache).

Для уточнення результатів відгук користувачів в кожному розрізі буде враховуватися з заданим коефіцієнтом від 1 до 10.

## 2.6 Розробка продукційної моделі для керування ліжком

Визначимо деякі правила, які згодом будуть використані в системі продукційних моделей. Виділимо з загальної множини параметрів сну підмножину параметрів, які можна віднести до передумов (сфер використання): температура в спальні і зовнішня, атмосферний тиск, різниця температур і тисків з минулим і майбутнім, опади.

Далі розглянемо одну з підмножин при найбільш нейтральних передумовах. Під нейтральними передумовами мається на увазі стан навколишнього середовища, характеристики якого наведені у формулі 9.

$$= \{ \text{пр} = \text{зов} \pm 5^\circ\text{C}; = 760 \pm 5 \text{ мм рт. ст.}; = ; \in (21,24];$$

(9)

У таблиці 2.3 представлений список імовірнісних продукційних моделей для описаної вище передумови. Перша частина таблиці описує керовані параметри, які система задасть для досягнення бажаного результату, також деякі значення можуть бути задані, як зафіксовані користувацькими уподобаннями. Друга ж частина таблиці являє собою імовірнісні значення досягнення певного результату (вказано в процентах).

Таблиця 2.3 – Правила продукційних моделей

№	Регульовані параметри										Цільові параметри (%)				
	t <sub>буд</sub>	H <sub>п</sub>	A <sub>п</sub>	S <sub>п</sub>	H <sub>с</sub>	S <sub>с</sub>	H <sub>н</sub>	A <sub>н</sub>	S <sub>н</sub>	T <sub>вн</sub>	SD	D	F	MP	HA
1	05-07	10	30	20	5	35	0		10	23	52	27	15	14	15
2	06-07	10		20	5		0	20	10		95	57	70	8	21
3	07-09	10	25	20		35		20	10		55	10	35	20	3
4	none	10	30	20	5	35	0	20		25	76	13	8	10	5
5	05-07		0		5	40	15	-10	30		70	37	30	15	41
6	06-07		0		5	30	15	-10	30	24	85	18	25	9	10
7	07-09		-5		5		15	-10	30		50	30	15	5	15
8	none		0		5	40	15	-10	30		96	60	77	11	22
9	05-07	5	0	35	10	30	0	-25		20	50	10	37	20	4
10	06-07	5	0	35	10	30	0	-25		22	84	15	12	10	5
11	07-09	5	5	35	10	30	0	-25			70	40	27	14	45
12	none	5	0	35	10	30	0	-25		20	85	22	25	11	10
13	05-07	10		10				-5		25	90	67	68	14	18
14	06-07	10		10				-5			53	10	34	13	6
15	07-09	10	-5	10				-5			80	16	4	9	5
16	none	10		10				-5		24	71	40	33	18	45



знаходження вхідних параметрів продукційної моделі будемо використовувати механізм зворотнього висновку.

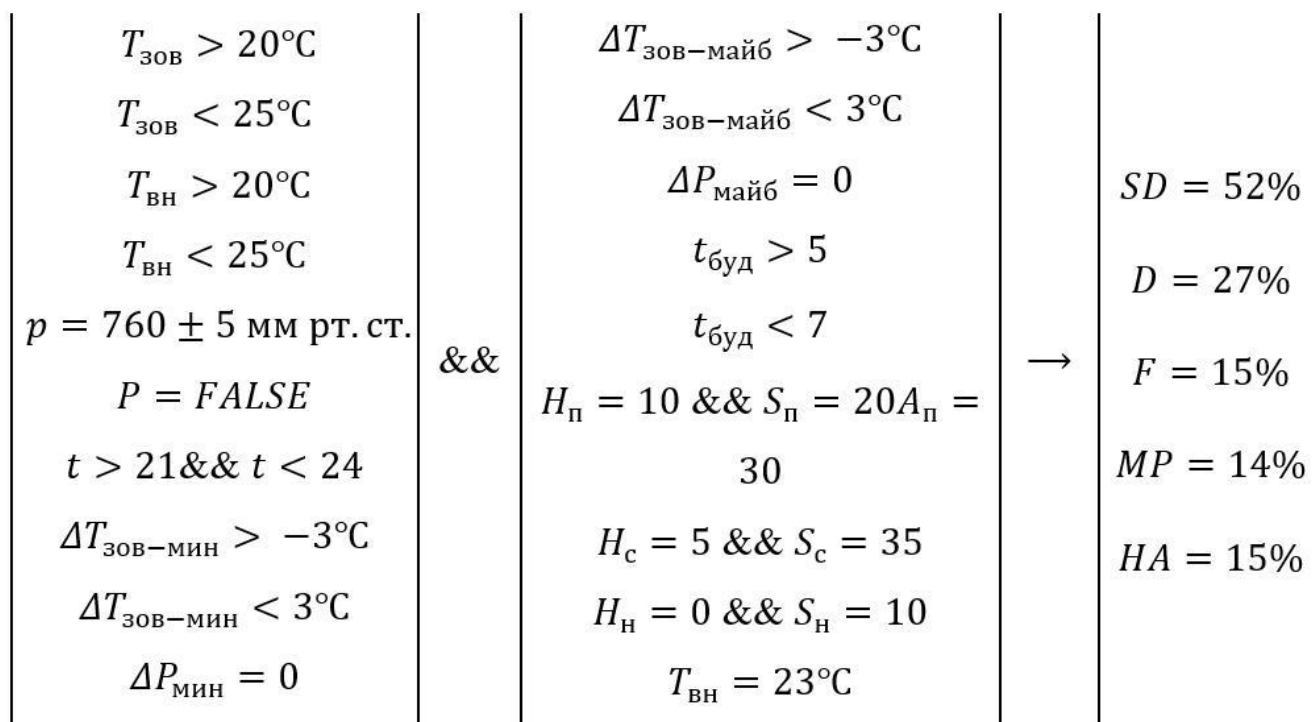


Рисунок 2.7 – Приклад правила в форматі продукційної моделі

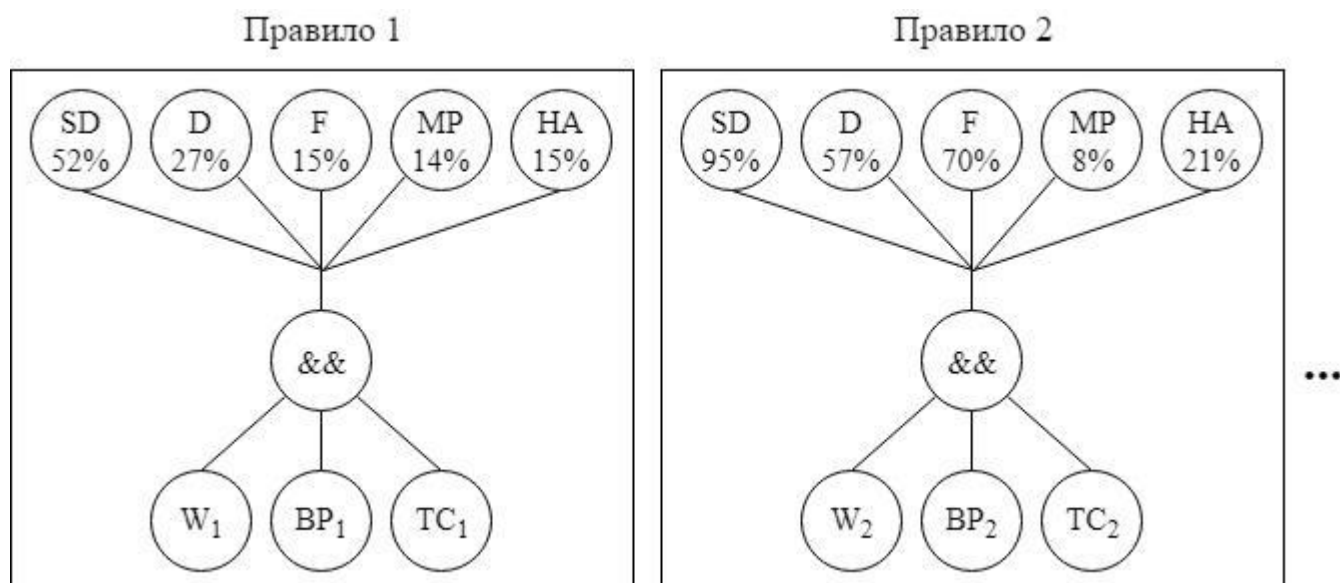


Рисунок 2.8 – Схематична структура системи продукцій

Використаємо схему, що наведена на рисунку вище, для наочного прикладу зворотнього висновку на предметній моделі.

Крок 1. Визначається гіпотеза “пріоритетний SD параметр” – глибина сну, щоб вночі менше реагувати на зовнішні подразники.

Крок 2. Знаходиться правило, із найбільшою вірогідністю оптимізованої глибини сну, на нашому прикладі це правило 2.

Крок 3. Досліджується можливість застосування Правил 2, тобто вирішується питання про те, чи може воно спрацювати. Для цього в робочій пам'яті повинні бути присутніми факти, що збігаються зі зразком цього правила.

Припустимо, що дані із датчиків не збігаються із визначеними у правилі  $TC_2$ , тобто Правило 2 не може спрацювати. Цей факт стає новою метою на наступному кроці виводу.

Крок 4. Знаходимо наступне правило, яке підходить по результуючому параметру. Повторюємо кроки 2-3 до тих пір, доки не знаходимо правило, вхідні параметри  $TC$  і  $W$  якого містять в собі реальні дані із дошки оголошень.

Припустимо, що параметри  $TC_1$  і  $W_1$  збігаються із вимогами із інформацією з датчиків системи.

Крок 5. Досліджується можливість застосування Правил 1. Воно спрацює, тому що в дошці оголошень присутній факт, що співпадає з його зразком. Шукані параметри  $BP$  будуть використані в налаштуваннях системи для досягнення бажаного результату.

Необхідно також врахувати відмінність між стандартизованими підходами до оцінки якості сну, представленими в правилах продукційних моделей вище, і ситуацією, що залежить від індивідуальних особливостей людей. Щоб привести систему до реальної оптимізації показників сну, потрібно ґрунтуватися безпосередньо на досвіді користувачів у використанні системи.

Тож, може бути використаний принцип пріоритетного вибору, який пов'язаний з введенням статичних або динамічних пріоритетів на продукції. Статичні пріоритети можуть формуватися на підставі відомостей про важливість

продукційних правил у цій проблемній області. Ці відомості, як правило, представляють собою інформацію, що надається експертами. Динамічні пріоритети виробляються в процесі функціонування системи продукцій і можуть відображати, наприклад, такий параметр, як час знаходження продукції в наборі активних продукцій. [15]

На підставі цього необхідно розбити етап випуску продукту на дві частини. Перша частина буде мати на увазі під собою роботу системи, засновану на стандартизованих правилах, наведених вище. Використовуватися система буде обмеженою групою відібраних користувачів, які, вносячи суб'єктивні дані про якість сну, регулюватимуть продукційні моделі і дозволять враховувати реальний людський фактор. Цей етап також може бути поєднаний з бета-тестом.

Друга ж частина стане безпосередньо випуском продукту в широке коло користувачів. На цьому етапі вже будуть використовуватися моделі, навчені працювати з реальними людьми. Варто враховувати, що, нехай перший етап і дозволив ввести велику організованість в правила вибору оптимального стану системи, він все ж не міг дозволити охопити всі варіанти індивідуалізації станів. Саме тому буде проводитися збір оцінок якості сну з користувачів і на другому етапі: крім індивідуалізації системи під конкретного користувача у вигляді середніх показників сну, з певною періодичністю буде проводитися редагування продукційних моделей на підставі найбільш популярних думок користувачів.

## 2.7 Розробка бази даних

Однією з найважливіших частин системи розумного ліжка є база даних, що зберігає в собі всю інформацію про користувачів, ліжка, їхні блоки і режими для комфортного сну, тому проектування бази даних є також дуже важливим етапом проектування системи.

В якості моделі для розроблюваної бази даних була обрана реляційна модель. Реляційна база даних являє собою набір таблиць (сутностей). Таблиці складаються з колонок і рядків (кортежів). В середині таблиць можуть бути визначені обмеження, між таблицями існують відношення. За допомогою SQL можна виконувати запити, які повертають набори даних, одержуваних з однієї або декількох таблиць. Перевагами реляційної моделі баз даних є:

- простота (в реляційній моделі всього одна інформаційна конструкція, яка формалізує табличне представлення даних, звичне для користувачів);
- теоретичне обґрунтування (наявність теоретично обґрунтованих методів нормалізації відносин дозволяє отримувати базу даних з заданими характеристиками);
- незалежність даних (коли необхідно змінити структуру реляційної бази даних, це, як правило, призводить до мінімальних змін в прикладних програмах).

До недоліків такої моделі відносяться:

- низька швидкість при виконанні операції з'єднання;
- велика витрата пам'яті для подання реляційної бази даних.

На рисунку 2.9 зображена схема бази даних. Розглянемо її детальніше.

Умовно усі таблиці бази даних можна логічно поділити на дві частини:

- частина ручного керування розумним ліжком;
- частина для зберігання інформації, пов'язаної із продукційними моделями.

До блоку керування юзером належать таблиці, які описують сутності:

- користувача;
- ліжка;
- блоку матрацу;
- пристрою для реєстрації та роботи з ліжком як предметом інтернету речей;
- збереженого стану ліжка.

Решту таблиць можна віднести до блоку роботи із продукційною моделлю.

Розглянемо кожен з них.



Таблиці RuleParameter та RuleResult несуть в собі інформацію про параметри, які були раніше досліджені. RuleParameter включає в себе як параметри передумов, наприклад, дані про погоду та часові кондиції, так і множину результуючих параметрів. У кожного параметра є своє ім'я, позначення, а також одиниця виміру. RuleResult використовує референс на результуючі параметри з таблиці RuleParameter та несе інформацію про вірогідність свого настання.

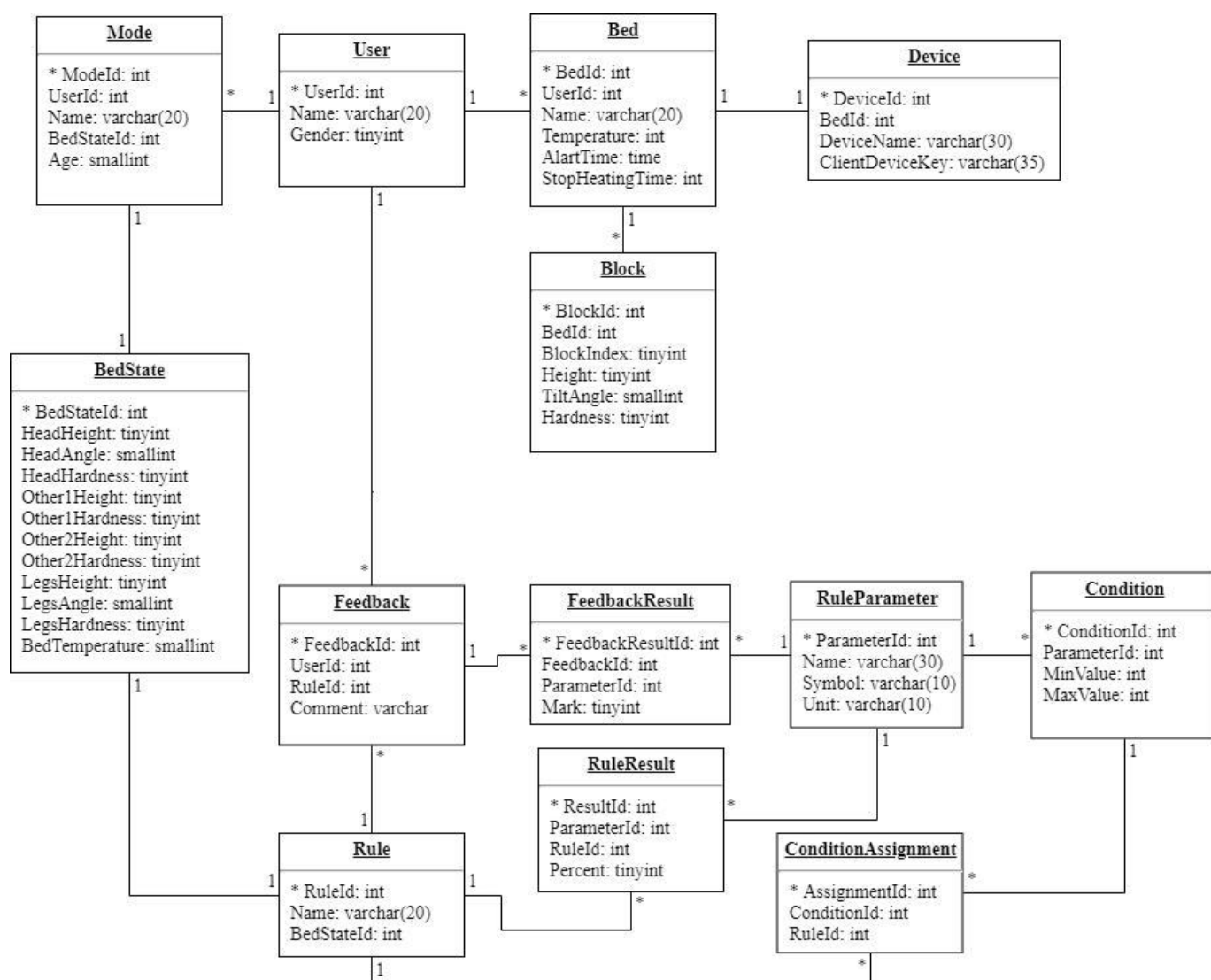


Рисунок 2.9 – Схема бази даних

Також таблиця RuleParameter використовується таблицею Condition, яка зберігає дані про різні умови правил продукційної моделі. В ній знаходяться дані про параметр і два значення, із якими реальне значення параметру буде

порівнюватися. За замовчуванням із мінімальним значенням реально буде порівнюватися виключно, із максимальним – включно.

Таблиця Rule володіє інформацією про сутність правил системи продукційної моделі. Кожне правило, окрім власного імені, має набір передумов, при яких правило буде активоване, інформацію про стан ліжка, яку потрібно буде застосувати системі, а також очікуваний результат сну користувача, якщо це правило буде активоване.

Слід зазначити, що одна й та сама умова може зустрічатися в декількох правилах, так як і правило буде складатися з багатьох умов, тож таблиця ConditionAssignment відповідає за зв'язок між цими двома таблицями.

Також маємо таблицю Feedback, за допомогою якої буде реалізована система навчання продукційної моделі. Користувач після активації одного з правил спав всю ніч із відповідними налаштуваннями ліжка, та вранці оцінив свій сон за результуючими параметрами. Як вже було сказано раніше, користувач оцінює кожен з критеріїв за 10-бальною шкалою, кожна оцінка відповідає конкретному значенню процентів, що зберігається в таблиці FeedbackResult. Тож саме за зрізами за таблицею FeedbackResult можна буде робити висновки щодо оптимізації системи правил продукційної моделі.

## 2.8 Розробка алгоритмів придбання знань та висновку для керування моделлю розумного ліжка

Для нормального функціонування системи необхідно розробити два основні алгоритми:

- логічного висновку на основі системи правил продукційної моделі;
- придбання нових знань на основі користувацького досвіду.

Ці два алгоритми ми розглянемо детальніше далі.

Як вже було сказано раніше, у продукційних моделях основним недоліком є висота ймовірність суперечливості знань при зависокій кількості правил. В проблемній області, що розглядається, через велику кількість визначених параметрів, що впливають на стан системи, очікується численна множина правил, тож, щоб оминати проблему їх суперечливості, необхідно привести правила до одного шаблону – шаблону опису усіх вхідних параметрів.

Це дасть нам впевненість в тому, що система не буде давати некоректні результати через суперечливість власних правил, але, натомість значно збільшить кількість правил, що використовується. Внаслідок цього рішення система правил буде містити в собі велику множину одиниць, прохід по якій може бути досить ресурсомістким. Тому доцільно ввести наступні покращення в алгоритм:

- фільтрувати цілу множину правил за даними дошки оголошень;
- додатково відсіювати правила, що не підходять за часом будильника, якщо той вказаний;
- сортувати підмножину правил, що залишилися, за результуючою ознакою.

Тільки після виконання дій, наведених вище, можна починати виконання алгоритму зворотнього висновку, не боячись нестачі ресурсів, схема виконання дій наведена на діаграмі послідовності (див. рис. 2.10).

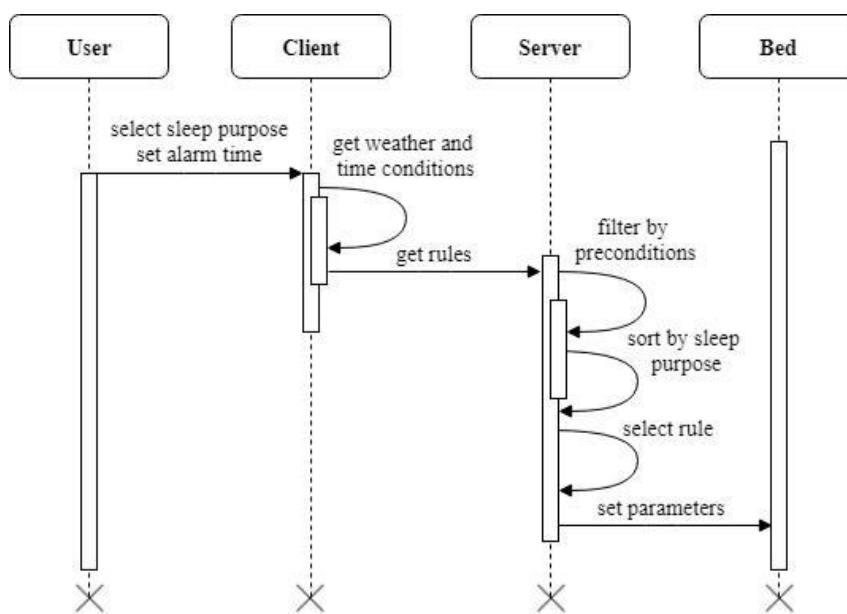


Рисунок 2.10 – Діаграма послідовності вибору параметрів

Тепер наведемо приклад вибору продукційної моделі на правилах, що визначені у таблиці 2.3, тобто для випадку, коли зовнішні умови є нейтральними. Припустимо, що користувачеві необхідно прокинутися о 6:30 і піти на важливі збори, що означає, що йому необхідно бути максимально бадьорим. В такому випадку ми повинні зробити кроки, що наведені нижче.

Крок 1. Отримуємо множину правил для поданих незмінних умов. Множина для вибору включає в себе 16 правил.

Крок 2. Визначаємо періоди, в який входить бажаний час будильника 6:30. Отримуємо два періоди: 5-7 та 6-7 години.

Крок 3. Фільтруємо множину правил за обраними періодами. Множина правил для вибору тепер має наступний вигляд:

$$= \{ 1, 2, 5, 6, 9, 10, 13, 14 \}; \quad (12)$$

У таблиці 2.4 наведені правила їхні параметри серед тих, серед яких необхідно зробити вибір.

Таблиця 2.4 – Правила для вибору, що підпадають під умову часу будильнику

№	F	H <sub>п</sub>	A <sub>п</sub>	S <sub>п</sub>	H <sub>с</sub>	S <sub>с</sub>	H <sub>н</sub>	A <sub>н</sub>	S <sub>н</sub>	T <sub>вн</sub>
P1	15	10	30	20	5	35	0		10	23
P2	70	10		20	5		0	20	10	
P5	30		0		5	40	15	-10	30	
P6	25		0		5	30	15	-10	30	24
P9	37	5	0	35	10	30	0	-25		20
P10	12	5	0	35	10	30	0	-25		22
P13	68	10		10				-5		25
P14	34	10		10				-5		

Крок 4. Згідно із умовами користувачу найбільш важливо бути бадьорим, тобто серед правил, що залишилися, необхідно обрати правило із мінімізованим . Обираємо правило P10.

Крок 5. Згідно із обраним правилом виставляємо усі параметри для ліжка. М'якість блоку для ніг, яка не задана у таблиці правил, залишаємо середньою для користувача.

Варто зауважити, що попередньо у ліжка задані стандартні параметри за замовчуванням, але пізніше, в ході роботи користувача із ліжком, стандартне значенні коригується тими значеннями, що юзер задав власноруч.

Таким чином відбувається вибір оптимального положення користувача під час сну для досягнення бажаних результатів.

Для досягнення постійної оптимізації системи правил продукційної моделі необхідно також спроектувати поведінку системи для отримання відгуків від користувачів.

Змоделюємо таку ситуацію:

- користувач за допомогою попереднього алгоритму обирає правило налаштувань ліжка, припустимо, P10.
- система активує правило та впроваджує відповідні налаштування, інформує користувача, що обране правило для оптимізації параметру бадьорості.
- вранці користувач отримує запит на опціональну оцінку якості свого сну за результуючими параметрами від одного до десяти.

При відносній простоті дій з точки зору користувача саме ця послідовність активностей надасть змогу системі оптимізувати систему продукцій наступним чином.

Коли набирається достатня кількість фідбеків щодо конкретного правила продукційної моделі, розпочинається процес оптимізації. Він включає в себе:

- збір статистики щодо фідбеків для конкретного правила;
- знаходження середніх значень за кожним результуючим параметром з фібдеків;

– отримання нових результуючих параметрів, що будуть записані у правило.

Нові параметри можна знаходити з використанням вагових коефіцієнтів за формулою:

$$n = 0.85 \cdot c + 0.15 \cdot \phi_{\text{сер}}, \quad (13)$$

$c$  – старі результуючі параметри;  
 $\phi_{\text{сер}}$  – середнє значення параметру з фідбеків.

Слід зазначити, що мале значення вагових коефіцієнтів при параметрах із відгуків дозволяє знизити вплив неадекватних оцінок користувачів, але це ж робить голоси усіх користувачів менш впливовими. Щоб уникнути цього, процес оновлення значень результуючих параметрів в правилах слід повторювати щомісячно.

## 3 РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ «РОЗУМНЕ ЛІЖКО»

### 3.1 Розробка архітектури системи

Під час планування та моделювання програмного продукту було визначено, що вся система повинні бути розділена на кілька складових:

- ознайомлювальний веб-сайт, який може переглядати будь-яка людина, а також виконувати функцію веб-клієнта для зареєстрованих користувачів або користувачів, які тільки купили свою ліжку;

- емулятор розумного ліжка, який є десктопним додатком під Windows OS (вибір припав на десктопні програми через аналогію предмету меблів – ним можна користуватися і без підключення до інтернету, але, як тільки воно з'являється, разом з ним приходить і велика кількість різної функціональності, як і належить предмету інтернету речей);

- сервер з базою даних, на якому будуть оброблятися запити ззовні, і зберігатися вся інформація про систему, її користувачів, їх налаштування та особисті уподобання.

Для наочного представлення побудуємо діаграму розгортання – статичну діаграму, яка служить для моделювання вузлів, які працюють, і артефактів, які розгорнуті на них, при цьому між артефактами і компонентами, які вони реалізують повинна бути встановлена залежність маніфестацій (див. рис. 3.1).

На рисунку 3.1 окрім основних модулів, що були наведені раніше, зображені шари кожного із цих модулів, вони описують основні види активності, що відбуваються на модулях. Так можна зрозуміти, що на сервері системи існують окремі шари, які відповідають за різні види діяльності:

- BeddestAPI приймає запити з клієнту;
- BeddestDAL (Data Access Layer) повністю відповідає за доступ до бази даних;
- Beddest Models містить класи сутностей, з якими працює систем;

- IoT Messages Transfer Module приймає і відправляє повідомлення на емулятор;
- Beddest Logic зв'язує в одну систему усі модулі, що були перелічені.

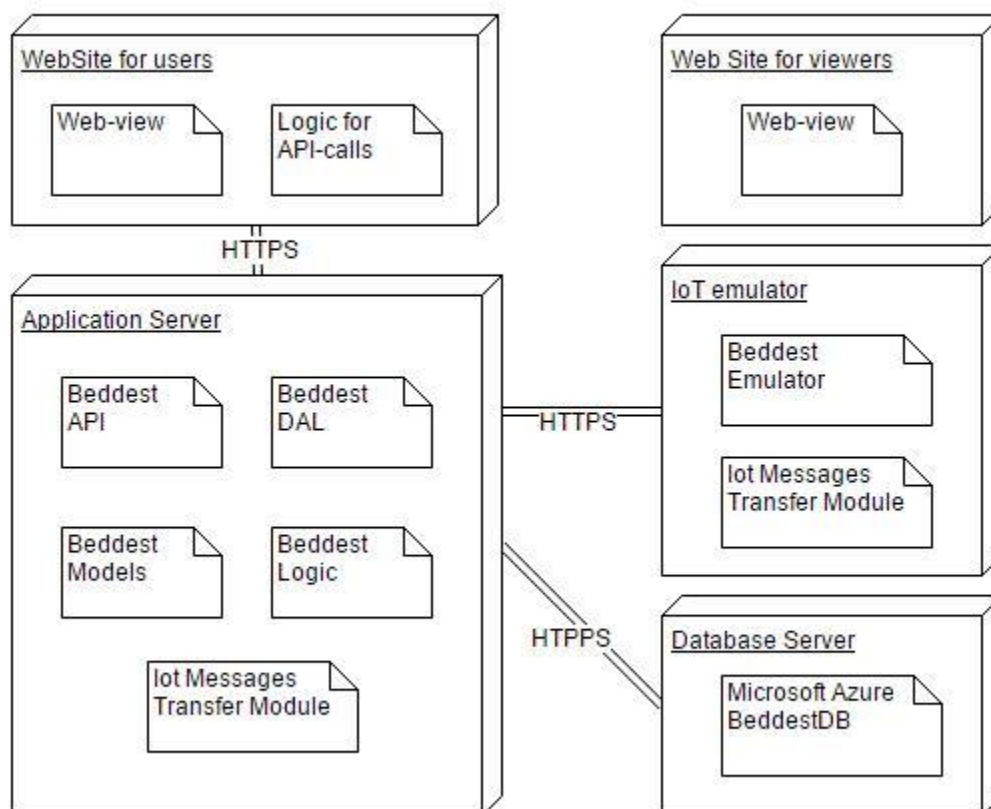


Рисунок 3.1 – Діаграма розгортання

Як можна побачити на діаграмі розгортання, деякі модулі також є багатошаровими. Наприклад, сервер додатку буде включати в себе окремі шари API, шар доступу до даних, моделей, логіки та шар зв'язку із IoT-хабом.

Однак, не можна сказати, що модуль веб-сайту напрямки зв'язується з модулем серверу – тільки окремі їх шари мають можливість спілкуватися між собою. Для наочного зображення таких зв'язків побудуємо діаграму компонентів – статичну структурну діаграму, яка показує розбиття системи на структурні компоненти, якими можуть бути файли, бібліотеки та модулі, і залежності між ними (див. рис. 3.2).



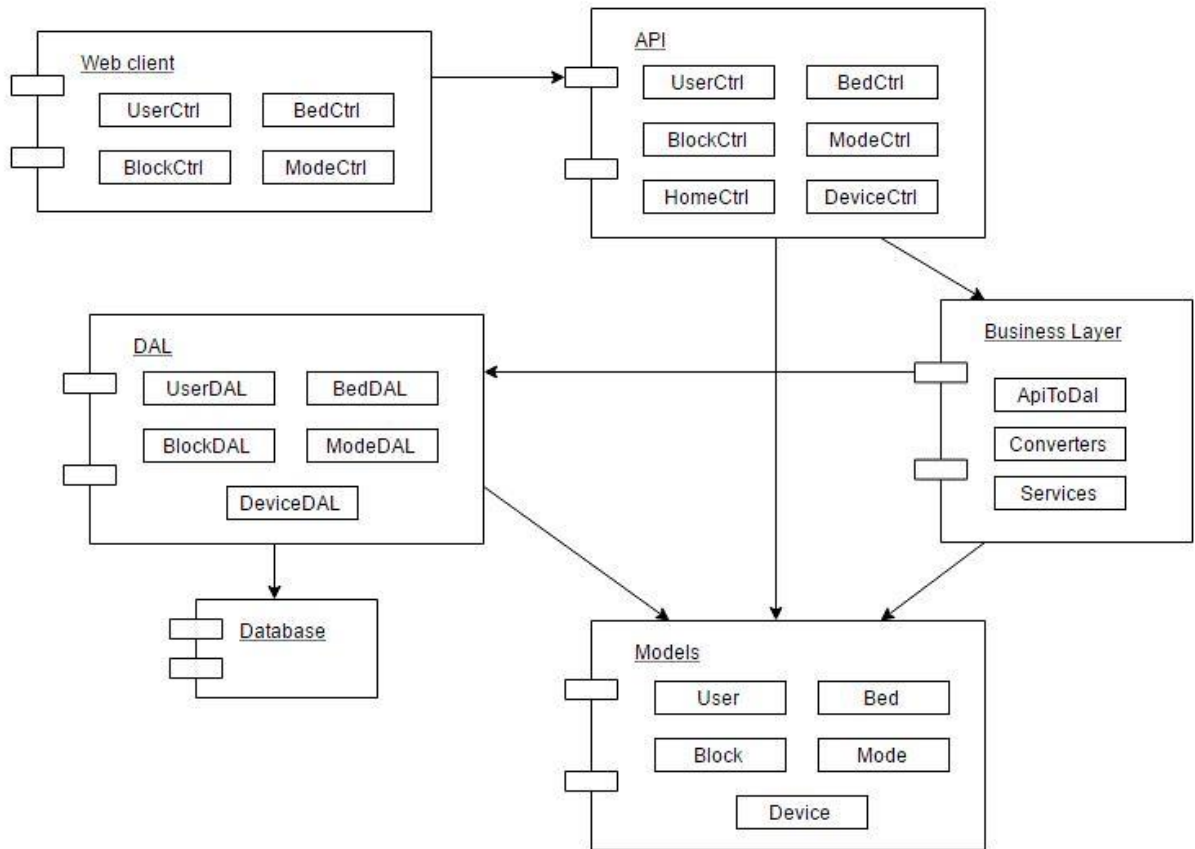


Рисунок 3.2 – Діаграма компонентів

На рисунку 3.2 на відміну від попереднього рисунку наочно показано, які з шарів усіх системних модулів взаємодіють між собою, а які не пов'язані між собою напряму.

### 3.2 Опис обраних технологій

Для розробки веб-сайту були використані кілька технологій. Сам сайт буде написаний з використанням HTML, CSS і JQuery. Розгортатися сайт буде завдяки Microsoft Azure [16] і доступ до нього буде відкритий для будь-якого користувача мережі інтернет. Також для створення сторінки був використаний AngularJS, що дозволяє пов'язувати введення текстової інформації або виведення області сторінки

з моделлю. Він також був використаний для реалізації зв'язку з сервером за допомогою аґах-запитів.

IoT-емулятор був реалізований за допомогою технологій WinForms та мови програмування C# платформи .NET [17] з використанням спеціальних бібліотек

Для координації даних, що відображаються на головному вікні емулятора, використовується Microsoft Azure IoT Hub, який дозволяє створити до 500 пристроїв-предметів інтернету речей, і AMQP (Advanced Message Queuing Protocol) протокол, а також HTTP-запити для роботи емулятора (але не власне пристрою!) для вибору пристрою, робота якого буде емулюватися.

Серверна частина – головний компонент системи Beddest. Система не буде повністю працездатною, якщо в ній буде відсутня хоча б один з перерахованих вище основних компонентів, однак без серверної частини система перестане існувати зовсім, залишаться лише розрізнені модулі.

Сервер є багаторівневим, має кілька окремих шарів, пов'язаних один з одним: шар загальнодоступного API, бізнес-шар, шар моделей, шар доступу до бази даних, а також доповнений шаром зв'язку з IoT Hub. Написані вони всі з використанням мови програмування C# платформи .NET, шар доступу до бази даних використовує ADO.NET, шар загальнодоступного API використовує ASP.NET Web API [18].

Одним із найголовніших шарів серверного модулю системи Beddest є шар доступу до бази даних, в якій зберігаються всі відомості про користувачів, стан їхніх ліжок, приватних або загальнодоступних режимів для них.

Сама база даних написана з використанням MsSQL. У такої бази даних буде багато переваг, наприклад:

- величезні обсяги даних;
- лінійна масштабованість;
- відмовостійкість;
- реляційність.

### 3.3 Опис фізичної моделі бази даних

Надзвичайно важливою частиною роботи системи є зберігання інформації в базі даних, на основі якої побудований весь функціонал системи. Сама база даних вже була спроектована в попередніх розділах, тепер наведемо деякі з SQL-скриптів для створення таблиць в базі даних.

Скрипт створення таблиці “Користувач”:

```
CREATE TABLE [dbo].[User] (
    [UserId] [int] IDENTITY(1,1) NOT
    NULL, [Name] [varchar](20) NOT NULL,
    CONSTRAINT [PK_User] PRIMARY KEY ([UserId]));
```

Скрипт створення таблиці “Ліжко”:

```
CREATE TABLE [dbo].[Bed] (
    [BedId] [int] IDENTITY(1,1) NOT NULL,
    [UserId] [int] NOT NULL,
    [Name] [varchar](20) NOT NULL,
    [Temperature] [int] NULL,
    [AlarmTime] [time](7) NULL,
    [StopHeatingTime] [int] NULL,
    CONSTRAINT [PK_Bed] PRIMARY KEY ([BedId]),
    CONSTRAINT [FK_Bed_User] FOREIGN KEY([UserId])
    REFERENCES [dbo].[User] ([UserId]);
```

Скрипт створення таблиці “Блок матрацу”:

```
CREATE TABLE [dbo].[Block] (
    [BlockId] [int] IDENTITY(1,1) NOT NULL,
    [BedId] [int] NOT NULL, [BlockIndex]
    [tinyint] NOT NULL, [Height] [tinyint]
    NOT NULL, [TiltAngle] [smallint] NOT
    NULL, [Hardness] [tinyint] NOT NULL,

    CONSTRAINT [PK_Block] PRIMARY KEY ([BlockId]),
```

```

CONSTRAINT [FK_Block_Bed] FOREIGN KEY([BedId])
REFERENCES [dbo].[Bed] ([BedId]);

```

Скрипт створення таблиці пристрою для реєстрації та роботи з ліжком як предметом інтернету речей:

```

CREATE TABLE [dbo].[Device] (
    [DeviceId] [int] IDENTITY(1,1) NOT
    NULL, [BedId] [int] NOT NULL,
    [DeviceName] [varchar](30) NOT NULL,
    [ClientDeviceKey] [varchar](35) NOT NULL,
    CONSTRAINT [PK_Device] PRIMARY KEY ([DeviceId]),
    CONSTRAINT [FK_Device_Bed] FOREIGN KEY([BedId])
    REFERENCES [dbo].[Bed] ([BedId]);

```

Розглянемо ближче метод зберігання передумов для системи правил продукційних моделей. Як вже розглядалося раніше, на етапі отримання правил необхідно одразу їх фільтрувати за передумовами. Для швидкого отримання даних із бази були спроектовані таблиці RuleParameter і Condition. Вони зберігають у собі інформацію щодо передумов таким чином, що за нею легко проводити запити.

Наприклад, із датчиків надійшла інформація про різницю температур між вулицею та приміщенням в 10 градусів. За умовами, що цей параметр в базі даних носить ідентифікаційний код 6, складемо запит, що отримує параметри ліжка для правил, що підходять за цими ознаками:

```

DECLARE @paramId int = 6;
DECLARE @tempDiff int = 10;
SELECT Rule.Name, BedState.*
FROM Rule
INNER JOIN ConditionAssignment AS ca
    ON ca.RuleId = Rule.RuleId
INNER JOIN Condition AS cond
    ON cond.ConditionId = ca.ConditionId
    AND cond.MinValue > @tempDiff
    AND cond.MaxValue <= @tempDiff
INNER JOIN RuleParameter AS param
    ON param.ParameterId = @paramId

```

```
INNER JOIN BedState
ON BedState.BedStateId = Rule.BedStateId
```

Таким чином вже на етапі вибірки даних із бази ми можемо впровадити фільтрацію, що значно пришвидшує процес їх обробки.

### 3.4 Інтерфейс і функціонал

Серед модулів системи тільки два мають user interface: веб-сайт та IoT-емулятор. Розглянемо їх обидва.

Для початку роботи із системою користувачеві необхідно зареєструватися у системі через веб-сайт (див. рис. 3.3), додати ліжко.

Обрати створене ліжко або ліжко, що вже існувало, для редагування (див. рис. 3.4) і почати редагувати (див. рис 3.5)

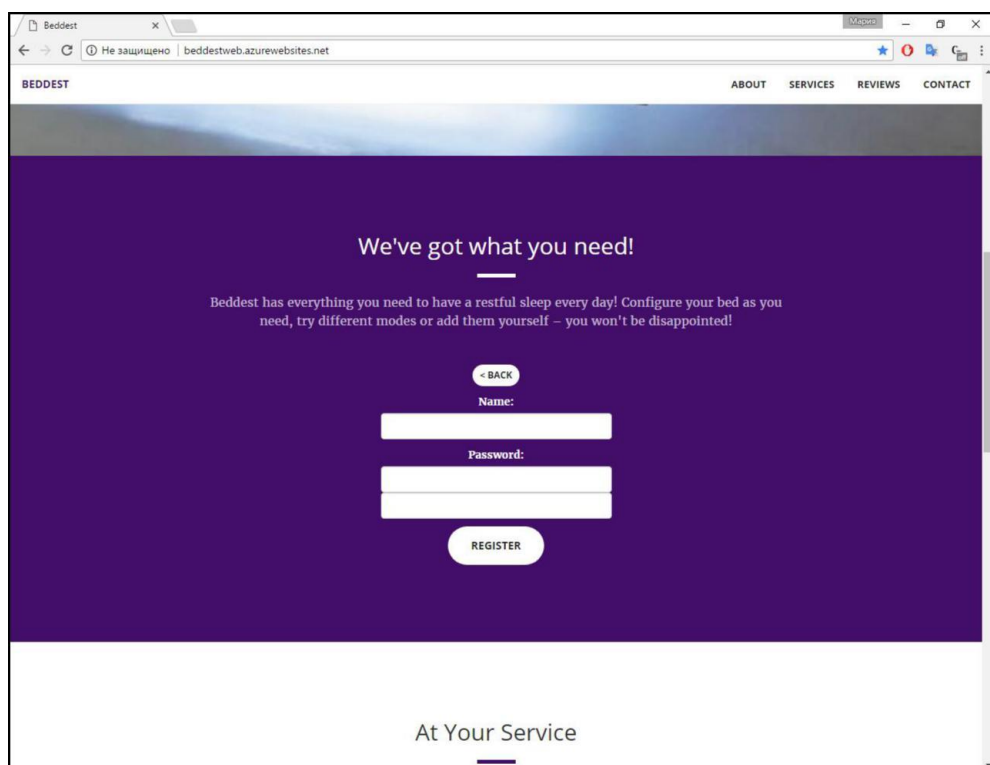


Рисунок 3.3 – Реєстрація в системі

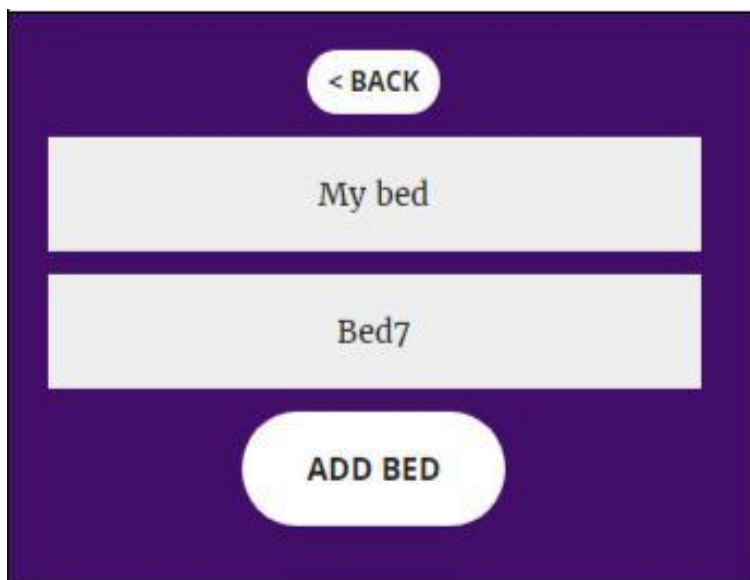


Рисунок 3.4 – Вікно обрання ліжка

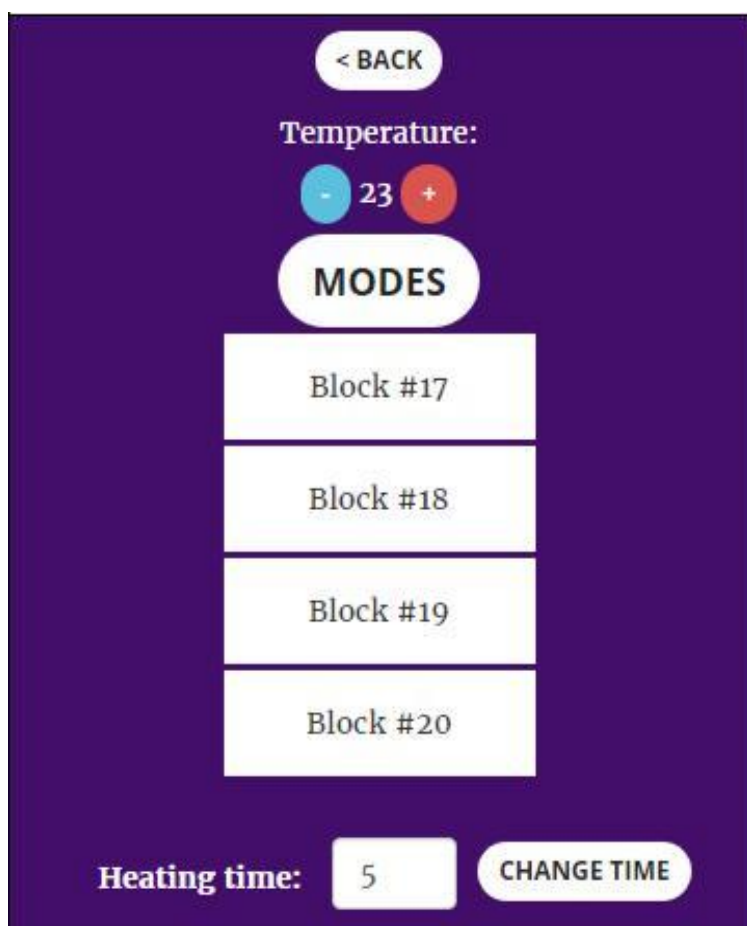


Рисунок 3.5 – Редагування ліжка

Для відображення результатів редагування користувач повинен застосувати десктопний IoT-емулятор.

Емулятор предмета інтернету речей – другий основний компонент системи, що розробляється.

Для збільшення юзабіліті інтерфейсу в головному вікні розумне ліжко показано двічі – в двох різних ракурсах: вид зверху для зручного вибору блоку, який користувач бажає відрегулювати, і вид збоку для наочної зміни положення блоків матраца (див. рис. 3.6).

Також на головному екрані з лівого боку схематично зображена людина, яка може зараз використовувати ліжко, та її параметри.

Параметри зросту та ваги отримується датчиками тиску на блоках ліжка, щоб автоматично обирати, із якого боку знаходиться блок подушки, а із якого – блок для ніг.

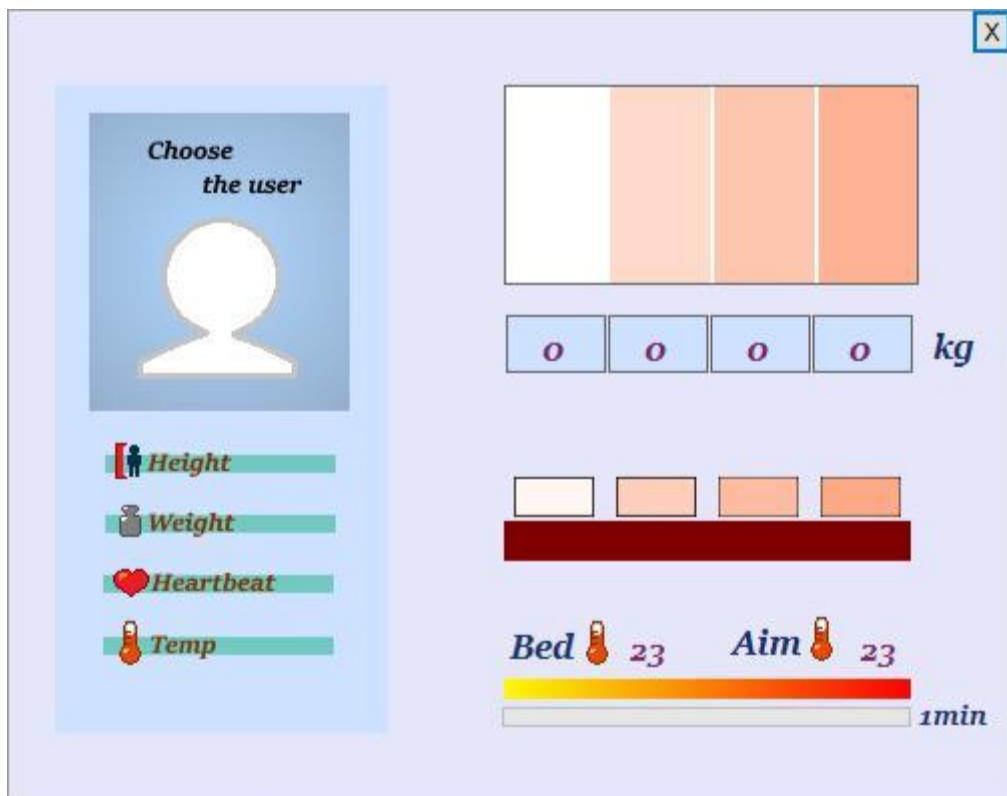


Рисунок 3.6 – Інтерфейс IoT емулятору

В реальній версії системи замість емулятору буде справжнє ліжко, придбане користувачем, але зараз для наочності користувачеві запропоновано вибрати ліжко, з яким він хоче працювати (див. рис. 3.7).

Коли ліжко для емулювання обране, користувач побачить головне вікно додатку, яке було зображене на рисунку 3.6.



Рисунок 3.7 – Вікно вибору ліжка для емулювання

Користувач може обрати людину, що ляже на ліжко, натиснувши на аватару панелі зліва (див. рис. 3.8). Коли людина обрана, вона буде відображена на панелі ліжка згори, а вага, яка розподіляється за блоками з'явиться на панелі вагів (див. рис. 3.9). Людину можна зсувати по ліжку за допомогою миші.

Якщо користувач змінить положення блоку на сайті (див. рис. 3.10), натиснувши на обраний блок, результат одразу відобразиться на головному вікні. (див. рис. 3.11). Також, якщо з рисунку 3.7 користувач змінить бажану температуру ліжка, це відобразиться в значенні "Air" нижньої правої панелі головного вікна емулятора, а значення "Bed" тієї ж панелі буде поступово змінюватись.

Якщо ж людина "встане" з ліжка, коли користувач натисне на хрестик біля аватару людини, то після певного часу, що зазначений в хвиликах у нижній правій панелі вікна цільова температура встановиться в температуру навколишнього простору, яке в реальному проекті вимірюється відповідними датчиками, а зараз за замовчуванням – 23С.





Рисунок 3.8 – Вибір людини, що ляже на ліжко

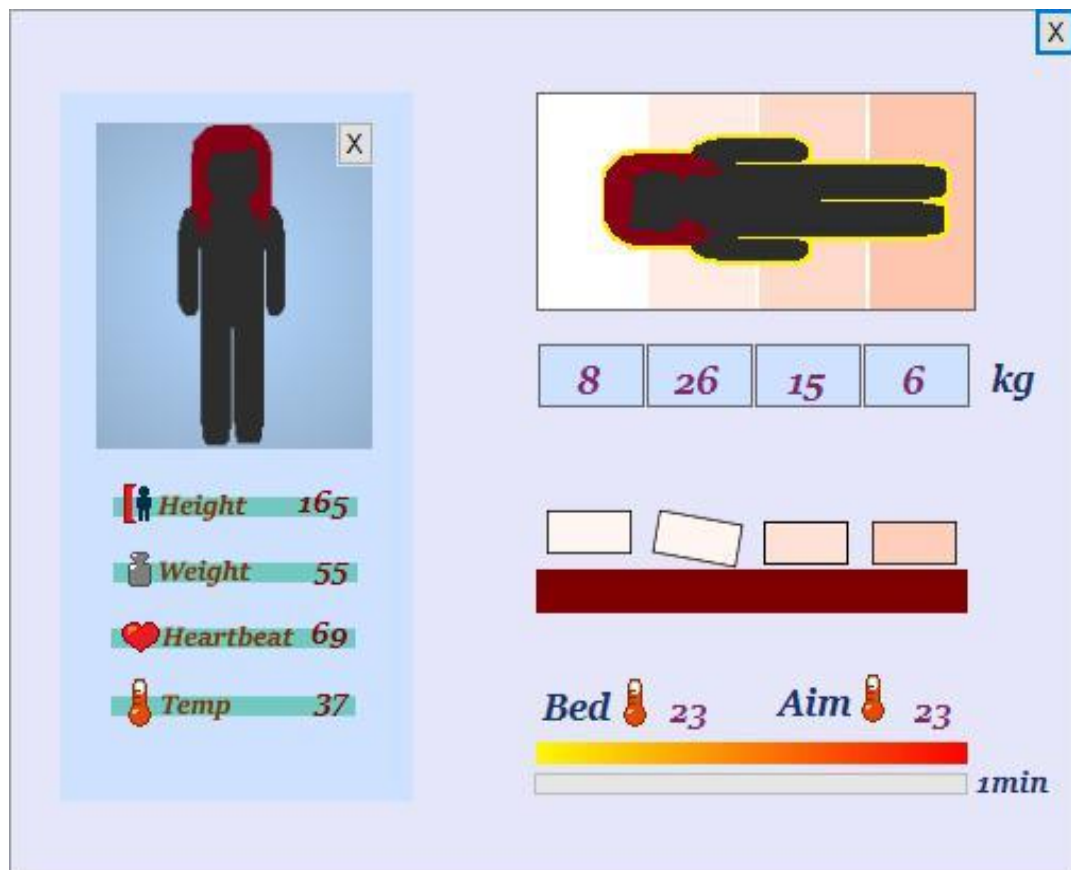


Рисунок 3.9 – Людина лягла на ліжко

Серверна частина напряму не використовується в проєкті і користувач не має змоги надіслати запит своїми руками, але спілкування між сайтом-керувачем і IoT-емулятором відбувається саме через нього.



Рисунок 3.10 – Регулювання блоку

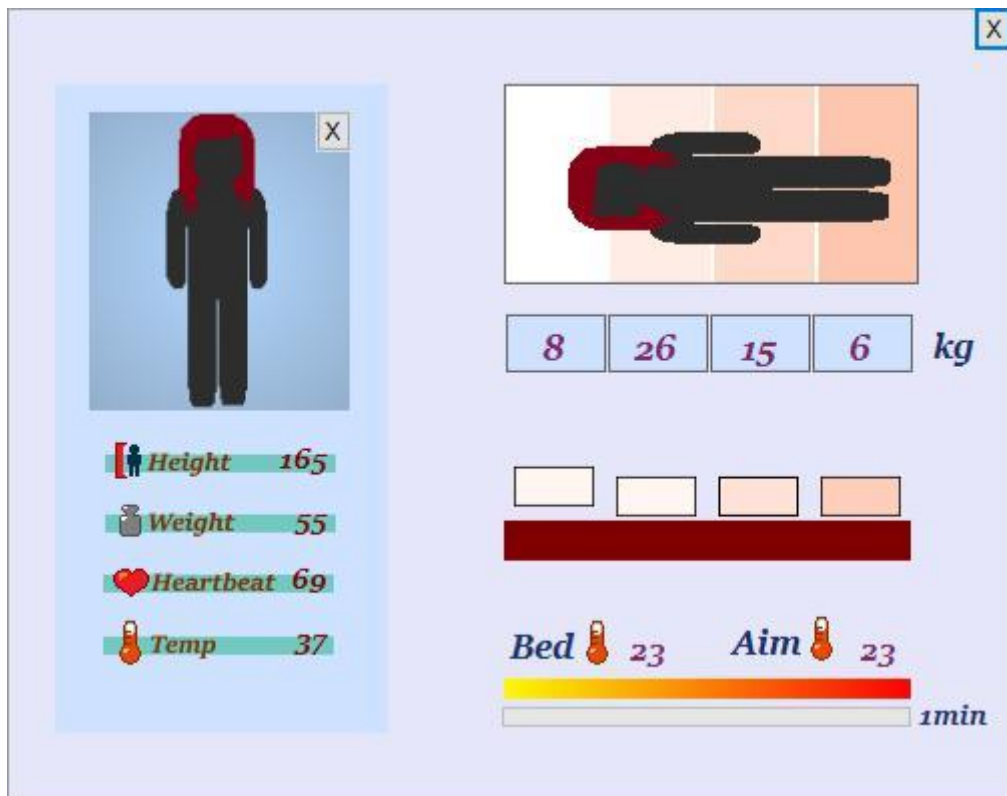


Рисунок 3.11 – Зміна положення блоку

Тож, приклади інтерфейсів тих двох модулів, що мають графічне відображення, були наведені, тепер наведемо також приклади програмного коду із усіх чотирьох модулів системи.

Далі наведений фрагмент коду веб-сайту для запиту логіну, а також наступного отримання даних про доступні ліжка. В цьому фрагменті були використані технології JS, AngularJS, JQuery.

```

$scope.login = function () {
    selectSection($('#loading'));
    login($scope.login.userName, $scope.login.password)
    .then(function (userData) {
        if (userData != null) {
            $scope.user.id = userData.Id;
            $scope.user.userName = userData.Name;
            getBeds($scope.user.id)
        }
        .then(function (bedsData) {
            $scope.beds = bedsData;
            selectSection($('#select-bed'));
        },
        function () { });
    }
    else {
        $scope.user.userName = "";
        $scope.user.password = "";
        alert("Wrong user name or password!");
    }
},
function () {
    console.log("failed");
})
};

apiAddress = 'http://beddest.azurewebsites.net/';
login = function (name, pass) {
    var deferred = $q.defer();
    $.getJSON(apiAddress + 'user/login?username=' + name
        + '&password=' + pass,
        function (data) {
            deferred.resolve(data);
        },
        function (data) {
            deferred.reject();
        });
    return deferred.promise;
};

```

Далі неведений фрагмент коду емулятора, за допомогою якого здійснюється передача повідомлень від пристрою до веб-серверу і навпаки. Код написаний на мові С# з використанням спеціальних бібліотек Microsoft Azure IoT.

```

public static DeviceClient deviceClient;
public static string IoTHubUri =
    "beddestIoT Hub.azure-devices.net";
public static async void SendDeviceToCloudMessagesAsync( string
    commandName, object objToSend)
{
    try
    {
        Command command = new Command(commandName,
            JsonConvert.SerializeObject(objToSend))
            ;
        string messageString =
            JsonConvert.SerializeObject(command) ;
        Message message = new Message(
            Encoding.ASCII.GetBytes(messageString)) ;

        await deviceClient.SendEventAsync(message) ;
    }
    catch { }
}

public static async void ReceiveCloudToDeviceAsync()
{
    if (deviceClient == null)
        deviceClient = DeviceClient.Create(IoTHubUri, new
            DeviceAuthenticationWithRegistrySymmetricKey(
                deviceId, deviceKey));

    while (true)
    {
        Message receivedMessage =
            await deviceClient.ReceiveAsync();
        if (receivedMessage == null) continue;

        await deviceClient.CompleteAsync(receivedMessage) ;

        string messageString = Encoding.ASCII
            .GetString(receivedMessage.GetBytes());
        Command command = JsonConvert

```

```

        .DeserializeObject<Command>(messageString);
        BlockDTO sdf = JsonConvert
            .DeserializeObject<BlockDTO>(command.Data.ToString());

        MessageProcessing.Process(command);
    }
}

```

Далі неведений фрагмент коду веб-серверу. Код написаний мовою С#. За допомогою цього коду здійснюється процедура додавання запису ліжка у базу даних, а також пов'язаних із ним блоків та пристрою.

```

public static async Task AddBed(int
    userId, int[] hardnessLevels)
{
    Bed newBed = new Bed(userId);
    int bedId = BedDataAccess.AddBed(newBed);

    int blocksCount = 4;
    Block[] blocks = new Block[blocksCount];
    for (int i = 0; i < blocksCount; i++)
        blocks[i] = new Block(bedId, hardnessLevels[i]);
    BlockDataAccess.AddBlocks(blocks);

    await AddDeviceAsync(bedId);
}

```

Нижче представлений також фрагмент коду із API модулю, в ньому налаштовується отримання інформації із девайсу на клієнт IoT Hub.

```

public static string iotHubD2cEndpoint = "messages/events";
public static EventHubClient eventHubClient;
public static List<EventHubReceiver> eventHubReceivers;

public static void StartReceiving()
{
    if (eventHubClient == null)
        CreateHubClient();
    var d2cPartitions = eventHubClient

```

```

        .GetRuntimeInformation().PartitionIds;

eventHubReceivers = new List<EventHubReceiver>();
foreach (string partition in d2cPartitions) {

    eventHubReceivers.Add(eventHubClient
        .GetDefaultConsumerGroup().CreateReceiver(partition,
            DateTime.UtcNow));
}
}

```

І також наведений алгоритм оптимізації значень результуючих параметрів правила, що був розглянутий раніше.

```

public void UpdateRuleResults(Rule rule)
{
    List<Feedback> ruleFeedbacks =
    FeedbackDataAccess.GetFeedbacksByRule(rule.RuleId);
    IEnumerable<FeedbackResult> feedbackResults =
        ruleFeedbacks
            .SelectMany(feedback => feedback.Results);

    double avgDepth = this.GetAverageMark(feedbackResults,
        ParameterEnum.SleepDepth);
    byte newDepth = this.GetWeightedMark(rule.Depth,
        avgDepth);

    double avgDrowsiness = this.GetAverageMark(feedbackResults,
        ParameterEnum.Drowsiness);
    byte newDrowsiness = this.GetWeightedMark(rule.Drowsiness,
        avgDrowsiness);

    double avgFatigue = this.GetAverageMark(feedbackResults,
        ParameterEnum.Fatigue);
    byte newFatigue = this.GetWeightedMark(rule.Fatigue,
        avgFatigue);
}

```

```

double avgMusclePain = this.GetAverageMark( feedbackResults,
    ParameterEnum.MusclePain);
byte newMusclePain = this.GetWeightedMark( rule.MusclePain,
    avgMusclePain);

double avgHeadache = this.GetAverageMark( feedbackResults,
    ParameterEnum.Headache);
byte newHeadache = this.GetWeightedMark( rule.Headache,
    avgHeadache);

FeedbackDataAccess.UpdateRuleResult(newDepth,
    newDrowsiness, newMusclePain, newHeadache);
}

private double GetAverageMark(
    IEnumerable<FeedbackResult> results,
    ParameterEnum parameter) => results
    .Where(result =>
        result.Parameter.ParameterId == (int)parameter)
        .Average(result => result.Mark);

private byte GetWeightedMark(byte
    oldValue, byte feedbackValue) =>
    (byte)(0.85 * oldValue + 0.15 * feedbackValue);

```

Увесь код був написаний із дотриманням загальноприйнятих правил щодо чистоти коду та забезпечення можливості легко його прочитати згодом. [19]

## ВИСНОВКИ

В ході атестаційної роботи була проаналізована та змодельована проблемна область здорового сну, досліджені основні компоненти сну, вплив на нього різних параметрів середі та користувача. Розроблена математична модель для опису керування розумним ліжком та системи сон-ліжко.

Було проведено дослідження методів оптимізації сну на основі штучного інтелекту, розглянуті можливі для використання моделі надання знань. В ході дослідження також були виділені параметри сну, передумови та параметри, на які розумне ліжко має прямий вплив, а також результуючі параметри сну, що повинні бути оптимізовані для досягнення бажаного результату.

Було детально розглянуто процес роботи з продукційними моделями, складено множину початкових даних для подальшого використання у системі.

На основі цього були спроектовані та реалізовані база даних для зберігання математичної моделі алгоритми придбання та висновку для керування моделлю розумного ліжка. Означені алгоритми були додатково оптимізовані для досягнення кращих показників продуктивності системи. Також була спроектована та реалізована система розумного ліжка.

Правильний сон допоможе людині підтримувати ритм свого життя, не дозволяти втомити впливати на неї, дозволить висипатися краще за одне і те ж час і зробить сон більш ефективним за допомогою використання правильної ліжка – розумного ліжка.

Із модифікованими алгоритмами вибору оптимального стану ліжка система дозволяє забезпечувати користувачеві правильний і повноцінний сон, який необхідний людині для підняття ефективності життєдіяльності, іншими словами:

- підтримувати все збільшується ритм сучасного життя, не відставати і не відчувати себе загнаним;
- не дозволяти втомити впливати на життя користувача;



- не давати накопичуватися стресам і негативним емоціям, адже відсутність правильного сну серйозно впливає і на психіку людини;

- дозволяти користувачеві кожен день відмінно висипатися;

- дозволяти користувачеві виглядати більш свіжим і відпочилим;

- підвищувати ефективність роботи користувача, що і призводить до підвищення рівня життя.

Тобто, щоб досягти усього, що перелічено вище, в ході роботи було зроблено:

- проведений аналіз та моделювання проблемної області керування розумними ліжками;

- досліджені моделі надання знань;

- розроблено математичну модель для опису керування розумним ліжком;

- спроектовано базу даних для зберігання даних з розробленої моделі;

- спроектовано архітектуру системи;

- розроблено алгоритми придбання знань та висновку для керування моделлю розумного ліжка;

- програмно реалізовано систему.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Тхостов А. Ш., Рассказова Е. И., Методы оценки субъективного качества сна и мыслей перед сном [Текст] // Москва — 2015.
2. Homeautomation [Електронний ресурс]. Режим доступу: [https://en.wikipedia.org/wiki/Home\\_automation](https://en.wikipedia.org/wiki/Home_automation) (дата звернення: 10.04.2019). – Загол. з екрану.
3. Balluga [Електронний ресурс]. Режим доступу: <https://www.kickstarter.com/projects/684490728> (дата звернення: 08.04.2019). – Загол. з екрану.
4. Sleep Number x12 [Електронний ресурс]. Режим доступу: <http://itc.ua/news/sleep-number-x12-umnaya-krovat-dlya-uluchsheniya-kachestva-sna/> (дата звернення: 08.04.2019). – Загол. з екрану.
5. Штучний інтелект [Електронний ресурс]. Режим доступу: <https://sites.google.com/site/upravlenieznaniami/tehnologii-upravleniaznaniami/iskusstvennyj-intellekt> (дата звернення: 14.04.2019). – Загол. з екрану.
6. Foldvary-Schaefer N., Grigg-Damberger M., Sleep: what we know, don't know, and need to know. [Текст] // J Clin Neurophysiol. — 2016. — Vol. 23, no. 1 (February). — P. 4-20
7. Дюк В., Емануель В., Інформаційні технології у медико-біологічних дослідженнях [Текст] // Київ. — 2013. — 528с.
8. Ізнак А.Ф., Ізнак О.В., Цикл сон-неспанья та зв'язок із депресією [Текст] // Науковий центр психічного здоров'я, Москва — 2016. — 2с.
9. Физиология процесса сна [Електронний ресурс]. Режим доступу: <https://anatomus.ru/physiology/son.html> (дата звернення: 10.04.2019) – Загол. з екрану.

10. Класифікація систем штучного інтелекту [Електронний ресурс]. Режим доступу: <https://studfiles.net/preview/2113977/page:3/> (дата звернення: 15.04.2019). – Загол. з екрану.
11. Крючковский В.В., Петров Э.Г., Соколова Н.А., Ходакова В.Е., Введение в нормативную теорию принятия решений [Текст] // Херсон — 2013. — 282 с.
12. Lewis H., Shapiro A., Dream reporting following abrupt and gradual awakenings from different types of sleep [Текст] // Journal of Personality and Social Psychology — 2016. — Vol. 2 — pp. 170-179.
13. Соболев И.М., Статников Р.Б., Выбор оптимальных параметров в задачах со многими критериями [Текст] // Москва — 2006. — Дрофа — 127с.
14. Moser D., Anderer P., Gruber G., Sleep Classification According to AASM and Rechtschaffen & Kales: Effects on Sleep Scoring Parameters [Текст] // Vienna — 2009. — Vol. 32, Issue 2 — pp 139-149.
15. Гребенник І.В., Методи підтримки прийняття рішень [Текст] // Харків : ХНУРЕ — 2010. — 127 с.
16. Microsoft Azure [Електронний ресурс]. Режим доступу: <https://portal.azure.com/> (дата звернення: 10.05.2019). – Загол. з екрану.
17. Троелсен, Э. Мова програмування C# 5.0 і платформа .NET 4.5, 6 вид. [Текст]/Пер, з англ. – М: ТОВ «И.Д. Вільямс», 2014. – 1312 с.
18. Learn about ASP.NET Web API [Електронний ресурс]. Режим доступу: <http://www.asp.net/web-api> (дата звернення: 10.05.2019). – Загол. з екрану.
19. Martin R. Clean code: a handbook of agile software craftsmanship. [Текст] // Upper Saddle River — New-Jersey, 2009, - 591 p.