

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
РАДІОЕЛЕКТРОНІКИ

Факультет Комп'ютерних наук
Кафедра Програмної інженерії

ЗВІТ
З ВИРОБНИЧОЇ ПРАКТИКИ
Місце проходження практики:
ТОВ «Солід Системз»

у період з "27" травня по "14" червня 2019 р.

Тема індивідуального завдання:
Програмна система для контролю якості сну

Студент. гр.ПЗП-16-7 _____ Милохін Д.О.
(підпис)

Керівник практики _____ ст.викл. Сковороднікова В.В.
(підпис)

Робота захищена з оцінкою _____ «__» _____ 2019

Комісія:

_____ ст. викл. Черепанова Ю.Ю.
(підпис)

_____ ст. викл. Широкопетлева М. С.
(підпис)

_____ ст. викл. Козел Н.Б.
(підпис)

Харків, 2019

РЕФЕРАТ

Звіт з виробничої практики: 30 с., 13 рис., 6 джерел.

АДМІНІСТРАТОР, ВОЛОГІСТЬ, СОН, ТЕМПЕРАТУРА, ЯКІСТЬ, TYPESCRIPT, WEB–САЙТ.

Об'єктом розробки є програмна система для контролю якості сну.

Метою є реалізація програмної системи, яка надає користувачам можливість контролювати якість свого сну, і таким чином отримувати корисний та цілющий сон.

Методи розробки базуються на використанні мови програмування TypeScript для серверної частини, на HTML5, CSS3, Angular для клієнтського веб-додатку, MYSQL у якості бази.

У результаті виконаної роботи здійснено реалізацію програмної системи, яка надає користувачам можливість контролювати якість свого сну, і таким чином отримувати корисний та цілющий сон.

ЗМІСТ

Вступ.....	4
1 Відомості про підприємство	5
2 Аналіз предметної області	7
3 Специфікація програмного продукту	10
3.1 Вступ	10
3.2 Загальний опис	12
3.3 Конкретні вимоги	14
4 Архітектура та проектування програмної системи	17
4.1 UML-проектування	17
4.2 Побудова ER-діаграми	19
4.3 Архітектура програмної системи	22
4.4 Кодування програмного продукту	24
Висновки	29
Перелік посилань	30

ВСТУП

Ні для кого не є секретом той факт, що сон дуже важливий для організму. Саме під час сну наш організм починає активно відновлюватися. Це допомагає нам не тільки відчувати себе краще, але і добре виглядати. Але не кожен знає яка саме температура та вологість повинні бути в кімнаті під час сну і не в кожного є час це визначати.

Дуже важливим є й те, в якій позі спить людина. Правильна поза для сну визначається багатьма факторами. Починаючи від віку та статі, і закінчуючи історією хвороби. Так, наприклад, для жінки необхідно спати сім або дев'ять годин, тоді як чоловікові потрібно шість-вісім. Людям з хронічною серцевою недостатністю і тим, хто переніс інфаркт, краще спати на правому боці, а людям, які страждають на мігрень краще спати на спині.

Система «Healthy Dream» особливо потрібна тим людям, які цінують свої сили та час, піклуються про своє здоров'я та хочуть максимально автоматизувати процес підготовки до сну.

Критерії успіху програмної системи: система дійсно допомагає користувачам покращити умови сну, кількість нових користувачів зростає, досягнут позитивний приток грошових коштів по даній системі протягом дванадцяти місяців, системою зацікавилися різного роду оздоровчі організації, лікарі й інші для розміщення реклами у системі.

Таким чином, метою роботи є розробка програмної системи, яка буде надавати користувачу можливість контролювати якість свого сну, за допомогою рекомендацій, щодо температури та вологості, які повинні бути в кімнаті під час сну. А також, користувач зможе вносити у систему інформацію про себе та історію хвороби, й на підставі цього отримувати рекомендації щодо того, які пози для сну є правильними та корисними.

1 ВІДОМОСТІ ПРО ПІДПРИЄМСТВО

Місце проходження практики, ТОВ «Солід Системз», це харківське підприємство, яке більш 15 років займається розробкою та супроводом програмного забезпечення. Підприємство співпрацює з замовниками з Америки та Європи.

ТОВ «Солід Системз» розпочало роботу як група, що надає послуги з адміністрування серверів. Сьогодні на підприємстві працює 285 співробітників у п'яти окремих підрозділах, які допомагають обслуговувати постійно зростаючу клієнтську базу з більш ніж 2 250 завершеними проектами.

Для клієнтів підприємство розробляє спеціальні веб-додатки, які варіюються від простих CMS-сайтів до складних SaaS-додатків, розміщених на серверах.

«Солід Системз» застосовує Agile практики для розробки інноваційних мобільних додатків для Android (як смартфонів і планшетів), так і iOS (iPhone, iPad) залежно від потреб клієнта.

Як постачальник послуг з розробки програмного забезпечення для стартапів, «Солід Системз» розробляє програми, починаючи від навчальних додатків до додатків для планування польотів.

Проект, на якому проходила практика – це веб-сайт, написаний на .NET Core та Angular. Замовник даного продукту – американська компанія.

Розробка програмного продукту виконується за технологією Scrum. Спринт триває два тижні. Кожен день проводяться мітинги команди для розуміння поточного стану запланованих на спринт задач. Також за необхідністю проводяться мітинги із замовником для уточнення вимог.

У якості власника продукту виступає замовник.

Команда, що працює над проектом, на якому проходила практика, є досить великою та складається з розробників, мануальних та автоматизованих тестувальників, дизайнера, верстальника, керівника проекту та бізнес-аналітика.

Команда поступово збільшується, тому що проект розвивається та потребує все більше ресурсів.

На проєкті ведеться документація, як бізнес-вимог, так і технічна. Написанням, збором та уточненням бізнес-вимог займається бізнес-аналітик на початку спринту шляхом спілкування з замовниками. Технічну документацію пишуть розробники.

У якості пропозиції щодо покращення функцій існуючого програмного забезпечення може бути перегляд кожні три місяця написаного коду, показників якості, які вимірюються у процесі виконання з метою підвищення якості вихідного коду, покращення продуктивності, стійкості та надійності програмного забезпечення.

2 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Даний продукт – програмна система, яка дозволяє користувачам автоматизувати процес підготовки до сну.

Програмна система буде використовуватися людьми, які слідкують за своїм здоров'ям та цінують свій час. Система буде надавати користувачам можливість швидко в будь-який час отримувати рекомендації, яка температура та вологість у кімнаті є припустимими, а також, яка поза для сну найліпшою. Абсолютних аналогів системи на даний момент немає. Є лише програми, наприклад Runtastic Sleep Better (див. рис. 1.1).



Рисунок 1.1 – Застосунок Runtastic Sleep Better

Такі програми слідкують за людиною під час сну, обирають найкращий проміжок часу для пробудження.

Але цей застосунок зовсім ніяк не пов'язан із функціоналом, який буде пропонувати програмна система «HealthyDream». Тобто загрози у вигляді конкретних конкурентів для майбутньої системи на даний момент не існує.

Потенційним користувачам необхідна програмна система, що допоможе не витрачати зайві зусилля та час на перевірку температури та вологості у кімнаті, вирішування, які показники є нормальними і що робити, щоб їх зробити комфортними для сну. Крім того майбутнім користувачам не знадобиться проходити зайве обстеження у лікаря, витрачати на це гроші та час, щоб дізнатися у якій позі та при яких умовах йому потрібно спати, щоб відчувати себе добре після сну. Інформація, яка надається користувачам у системі, буде проаналізована провідними фахівцями з області людської фізіології, щоб користувачі отримували тільки якісні та правдиві дані та рекомендації. А також не менш важливим є те, що користувачі зможуть публікувати інформаційні пости, таким чином обмінюючись корисним досвідом з іншими. Такі пости можуть публікуватися й справжніми лікарями, адже вони теж зможуть зареєструватися у системі, як звичайні користувачі, й таким чином будуть набувати популярності, що для «Healthy Dream» буде неабияк добре та вигідно.

Система є масивною, тобто її потрібно буде обов'язково підтримувати. Також можуть виникнути проблеми із налаштування та технічним обслуговуванням обладнання – а саме, датчиків вологості та температури. Ще важливими ризиками є невиконання запланованих робіт, нестача грошових коштів на етапу виходу на проектну потужність, неотримання очікуваного доходу від реалізації проекту.

На базі аналізу інформаційних потреб користувачів, таких як адміністратор та звичайний користувач, були сформовані вимоги для кожного.

Адміністратор:

- а) реєстрація в програмній системі;
- б) авторизація в програмній системі;
- в) перегляд постів;
- г) підтвердити або відхилити пост;
- д) перегляд особистої інформації;

- е) редагування особистої інформації;
- ж) змінити пароль;
- з) видалити користувача;
- і) вихід із системи.

Інформаційні потреби звичайного користувача:

- а) реєстрація в програмній системі;
- б) авторизація в програмній системі;
- в) перегляд постів;
- г) додавання та редагування посту;
- д) змінити пароль;
- е) перегляд рекомендацій;
- ж) вихід із системи.

На основі аналізу предметної області і інформаційних потреб можна виділити наступні об'єкти системи і їх атрибути:

- а) користувач: унікальний ідентифікатор користувача, e-mail, хеш пароля, унікальний ідентифікатор персональної рекомендації, стать;
- б) хвороба: унікальний ідентифікатор хвороби, назва, унікальний ідентифікатор рекомендації;
- в) рекомендація: унікальний ідентифікатор рекомендації, поза, додатки;
- г) персональна рекомендація: унікальний ідентифікатор персональної рекомендації, температура, вологість, поза, додатки.

3 СПЕЦИФІКАЦІЯ ПРОГРАМНОГО ПРОДУКТУ

3.1 Вступ

Ні для кого не є секретом той факт, що сон дуже важливий для організму. Саме під час сну наш організм починає активно відновлюватися. Це допомагає нам не тільки відчувати себе краще, але і добре виглядати. Але не кожен знає яка саме температура та вологість повинні бути в кімнаті під час сну і не в кожного є час це визначати.

Нестача сну негативно впливає на нашу пам'ять. Під час сну наш мозок обробляє одержану за ніч інформацію. Якщо ви виділяєте на сон мало часу, то мозок просто не встигає справлятися зі своїм завданням, що веде до того, що ми не тільки не можемо засвоїти нову інформацію, але і забуваємо стару, оскільки мозок «перекриває до неї доступ». Відсутність повноцінного відпочинку вночі призводить до додаткового надолуження відпочинку вдень, що виражається в апатії, ліні, сонливості і просто денному сні, що дуже шкідливо для здоров'я.

Коротко про наслідки через відсутність/порушення сну:

- а) виникнення раку: ризик захворіти на рак молочної залози в два рази вищий для тих, хто лягає спати надто пізно;
- б) серцевий напад: ризик серцевого нападу збільшується на 100% у тих, хто спить менше 7 годин на добу;
- в) ожиріння: замість того, щоб дивитися телевізор, спробуйте спати на 1 годину більше, і побачите, як втрачаєте 6,3 кг на рік;
- г) ризик смерті: відсутність або порушення сну означає, що ви на 20% маєте більше шансів померти в 20 років.

Крім того дуже важливим є й те, в якій позі спить людина. Правильна поза для сну визначається багатьма факторами. Починаючи від віку та статі, і закінчуючи історією хвороби. Так, наприклад людям з хронічною серцевою недостатністю і тим, хто переніс інфаркт, краще спати на правому боці. Як відомо, у сні мозок людини продовжує працювати. У зв'язку з цим він потребує постійного

хорошого кровопостачання. Надходження крові, збагаченої киснем, в головний мозок здійснюється двома внутрішніми сонними і двома хребетними артеріями. При неправильній позі відбувається здавлювання цих судин, внаслідок чого порушується надходження крові в мозок. Перетиснення під час сну як обох, так і однієї хребетної артерії становить небезпеку. Коли виникає порушення кровопостачання одночасно двома хребетними артеріями, розвивається гіпоксія (киснєве голодування) клітин мозку. Навіть якщо здавлюється тільки одна артерія, друга при цьому може виявитися нездатною компенсувати потреби головного мозку. Адже нерідко зустрічається, коли одна артерія має більш вузький діаметр або її просвіт звужений за рахунок атеросклеротичної бляшки. Така артерія явно не зможе працювати «за двох».

Таким чином, здавлювання сонних і хребетних артерій під час сну становить загрозу для головного мозку. Отже, необхідно подбати про те, щоб у сні артерії не пережималися і кровотік не порушувався.

Не треба забувати й про вологість у кімнаті під час сну. Здоровий мікроклімат сьогодні в тренді. Тому важливо знати, яка вологість повітря повинна бути в кімнаті. Якими способами підвищити або знизити рівень? А також до яких захворювань може призвести неправильні кліматичні умови в житловому приміщенні?

Ступінь вологості залежить від пори року, погодних умов та регіонального клімату – ці фактори визначають оптимальне повітряне середовище як окремого регіону, так і конкретного приміщення. Взимку вологість помітно знижується, а влітку стає вище. Від якості повітря залежить здоров'я і самопочуття людей.

Таким чином було вирішено створити програмну систему, яка буде надавати користувачу можливість контролювати якість свого сну. Користувач зможе вносити у систему інформацію про себе та історію хвороби, й на підставі цього отримувати рекомендації щодо того, які пози для сну, температура та вологість є правильними та корисними.

3.2 Загальний опис

Дана програмна система буде використовуватися людьми, які слідкують за своїм здоров'ям та цінують свій час. Користувач буде зацікавлений тим, що зможе вносити та редагувати інформацію про себе та свій стан здоров'я, й на підставі цього буде отримувати рекомендації щодо того, яка температура, вологість та поза для сну будуть найбільш підходящими.

При проектуванні вимог до даної програмної системи були визначені наступні критерії успіху:

а) система дійсно допомагає користувачам покращити умови сну, як наслідок кількість користувачів зростає;

б) досягти позитивного потоку грошових коштів по даній системі протягом 12 місяців;

в) системою зацікавилися різного роду оздоровчі організації, лікарі і т.д. для розміщення реклами у системі;

г) кількість нових користувачів зростає.

Крім того були визначені бізнес-риски, з якими можна зіткнутися:

а) вартість технічної реалізації;

б) невиконання запланованих робіт;

в) нестача грошових коштів на етапі виходу на проектну потужність;

г) активний розвиток веб-технологій (зайві витрати на повторну розробку з новими технологіями);

д) конкуренція з іншими сервісами.

Метою програмної системи є надання користувачам можливість контролювати якість свого сну, і таким чином отримувати корисний та цілющий сон. «Healthy Dream» особливо потрібен тим людям, які цінують свої сили та час, хто піклується про своє здоров'я та хоче максимально автоматизувати процес підготовки до сну.

При розробці системи приймаються до уваги наступні допущення:

- а) користувачі даної програмної системи мають доступ до мережі Інтернет;
- б) можливе відхилення від точних значень температури та вологості, так як користувач може ввести неправильні дані.

На рисунку 3.1 зображені заінтересовані у даній програмній системі сторони:

<u>Stakeholder</u>	<u>Major Value</u>	<u>Major Interests</u>	<u>Constraints</u>
Звичайні користувачі	Швидке отримання інформації про поточну температуру, вологість у приміщенні. Отримання рекомендацій щодо правильних умов для сну.	Отримання корисного та цілющого сну. Покращення самопочуття.	Важкість налаштування зв'язку між датчиками та пристроєм. Датчики можуть бути пошкоджені.
Оздоровчі організації	Рекламування організації	Залучення нових клієнтів	Користувачам може не подобатися реклама

Рисунок 3.1 – Стейкхолдери програмної системи

Стейкхолдер (англ. Stakeholder), зацікавлена сторона, причетна сторона – фізична особа або організація, що має права, частку, вимоги або інтереси щодо системи або її властивостей, що задовольняють їх потребам і очікуванням.

У системної інженерії стейкхолдери розглядаються в контексті процесу прийняття рішень як фізичні особи або організації, що залежать від результатів прийнятих рішень. Розуміння того, хто є стейкхолдерами по відношенню до прийнятих рішень, має бути встановлено заздалегідь. Зацікавлені сторони по відношенню до організації можуть бути як зовнішніми, так і внутрішніми. До зовнішніх зацікавлених сторін відносяться клієнти, кредитори, постачальники,

представники органів державної та місцевої влади, громадські організації, засоби масової інформації, а серед внутрішніх зацікавлених сторін називають власників, вище керівництво, робітників

На рисунку 3.2 наглядно зображено пріоритети проекту.

<u>Dimension</u>	<u>Constraint (state limits)</u>	<u>Degree of Freedom (state allowable range)</u>
<u>Features</u>		70-80% високо пріоритетних вимог мають бути реалізовані у версії 1.0
<u>Quality</u>	90% користувацьких перевірочних тестів мають бути для версії 1.0	
<u>Cost</u>		До 15% перевищення по бюджету можливі без розгляду куратора проекту

Рисунок 3.2 – Пріоритети проекту

Як можна побачити пріоритети розглядаються відносно версії програмної системи.

Для проектування програмної системи була обрани наступні операційні середовища:

- а) серверна частина написана на мові програмування Node.JS;
- б) для зберігання даних про користувачів використовується СУБД MySQL;
- в) клієнтський веб-застосунок написаний за допомогою HTML, CSS, JS, Angular.

3.3 Конкретні вимоги

Функціональні вимоги – це вимоги до програмного забезпечення, які описують внутрішню роботу системи, її поведінку: калькулювання даних,

маніпулювання даними, обробку даних та інші специфічні функції, які має виконувати система. Розробники ПЗ не реалізують бізнес-вимоги або варіанти використання. Вони реалізують функціональні вимоги, визначені фрагменти поведінки системи, що дозволяють клієнтам виконувати варіанти використання і виконувати свої завдання [6, ст.189].

Далі наведено списки того, що кожна з частин повинна виконувати:

а) веб-сайт;

- 1) реєстрація в програмній системі;
- 2) авторизація;
- 3) перегляд інформації користувача;
- 4) редагування інформації користувача;
- 5) додавання та редагування посту;
- 6) підтвердження або відхилення посту (адміністратор);
- 7) видалення користувача (адміністратор);
- 8) перегляд персональних рекомендацій;
- 9) локалізація: українська та англійська мови;

б) сервер;

- 1) додавання та видалення користувача;
- 2) перевірка даних користувача для авторизації;
- 3) надання інформації про користувача;
- 4) змінення інформації про користувача;
- 5) додавання нового поста;
- 6) внесення змін до посту;
- 7) підтвердження або відхилення посту;
- 8) надання рекомендацій;
- 9) формування персональних рекомендацій.

В системі можна виділити п'ять сутностей: користувач (унікальний ідентифікатор користувача, e-mail, хеш пароля, унікальний ідентифікатор персональної рекомендації, стаття), хвороба (унікальний ідентифікатор хвороби,

назва, унікальний ідентифікатор рекомендації), рекомендація(унікальний ідентифікатор рекомендації, поза, додатки) та персональна рекомендація (унікальний ідентифікатор персональної рекомендації, температура, вологість, поза, додатки).

Система, після того, як користувач зареєструється, заповнить загальну інформацію про себе (вік, стать і т.д.) та надасть історію своєї хвороби, буде аналізувати отриману інформацію та у якості результати надавати рекомендації, щодо того, яка температура, вологість та поза будуть ідеальними для сну та чому саме так. Користувач буде отримувати підказки щодо поз для сну. Наприклад, якщо користувачу було рекомендовано спати на спині, але ця поза може викликати біль у попереку, тоді система надішле підказку «Підкладіть подушку під коліна – це допоможе вашій спині зберегти правильне положення під час сну».

У наступних релізах системи «Healthy Dream» планується додати наступний функціонал:

- а) можливість реєстрації окремої групи користувачів – лікарів;
- б) буде створений форум, де будуть обговорюватися актуальні теми, питання;
- в) група «Лікарі» зможуть додавати статті, які потім зможуть переглядати звичайні користувачі.

Вимоги до зберігання даних: дані мають зберігатись у базі даних SQL. Час зберігання необмежений.

Вимоги до інтерфейсу: інтерфейс користувача повинен бути збудований так, щоб він підтримував коректне та стабільне функціонування у браузерях Microsoft Edge (усі версії), Chrome (починаючи з версії 52 включно), Firefox (починаючи з версії 50.0 включно), Opera (починаючи з версії 35.0 включно). Роздільна здатність екрану, на якому буде відображатися веб-сайт медичної системи, повинна бути не менш ніж 1024x768. Інтерфейс має бути інтуїтивно зрозумілим та легким у використанні.

Вимоги до безпеки: конфіденційні дані користувача, такі як його email та пароль, повинні зберігатися у базі даних у зашифрованому виді.

4 АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

4.1 UML-моделювання

Уніфікована мова моделювання (UML) – це сімейство графічних нотацій, в основі якої лежить єдина метамодель. За допомогою UML можна легко і наглядно зобразити складний процес чи алгоритм.

Опис функціональної структури надано у вигляді Use Case (див. рис. 4.1).

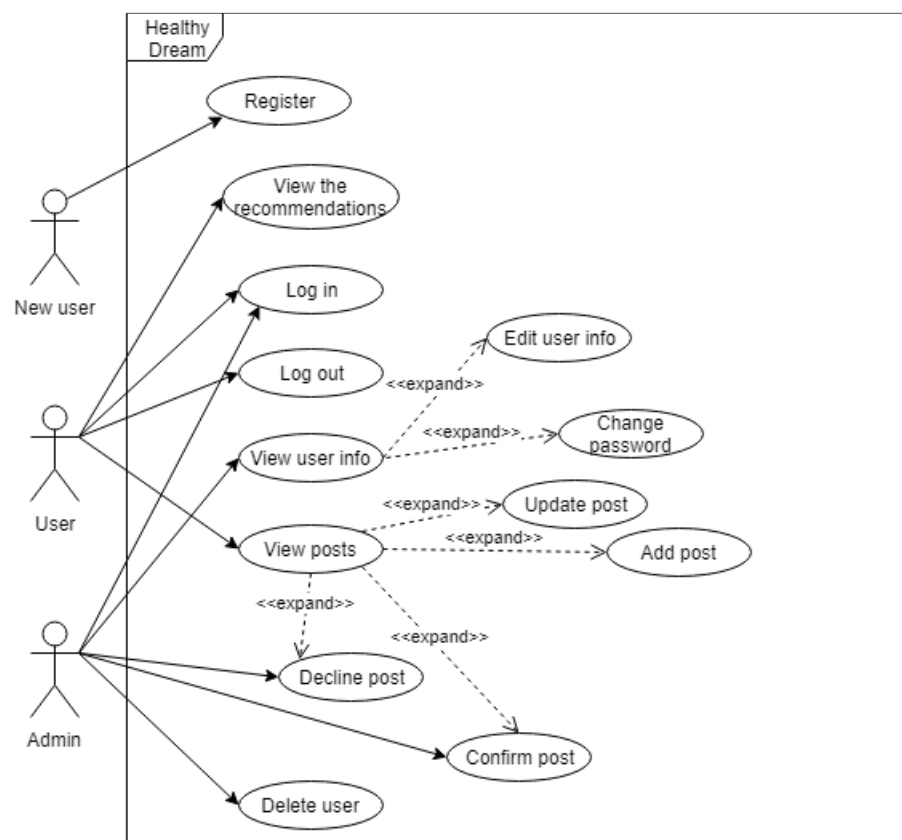


Рисунок 4.1 – Діаграма варіантів використання

Діаграма прецедентів — в UML, діаграма, на якій зображено відношення між акторами та прецедентами в системі. Суть даної діаграми полягає в наступному: проектувана система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант

використання використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором. При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

Нижче зображено діаграму компонентів (див. рис. 4.2) – статичну структурну діаграму, яка показує розбиття програмної системи на структурні компоненти та зв'язок (залежності) між компонентами.

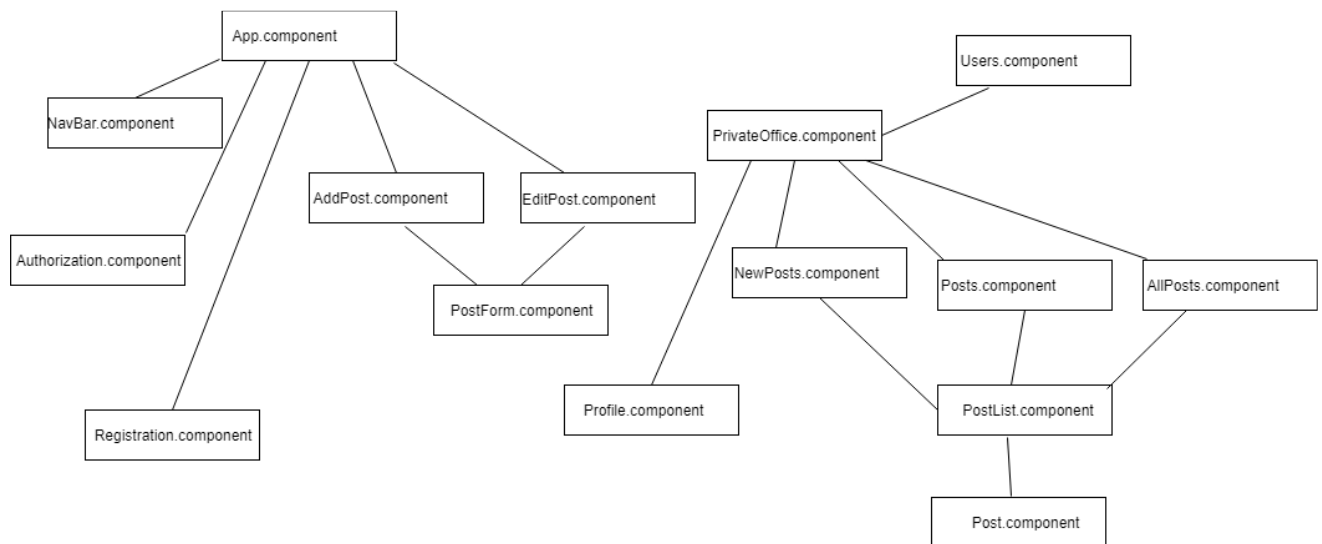


Рисунок 4.2 – Діаграма компонентів

Залежність показує, що один компонент надає сервіс, необхідний іншому компоненту. Залежність зображується стрілкою від одного компонента або до іншого. Діаграма показує відповідність класів реалізованим компонентам. Отже, вона потрібна там, де починається генерація коду.

На рисунку 4.3 наведено діаграму розгортання (Deployment diagram). На ній зображено працюючі вузли та розгорнуті в них артефакти.

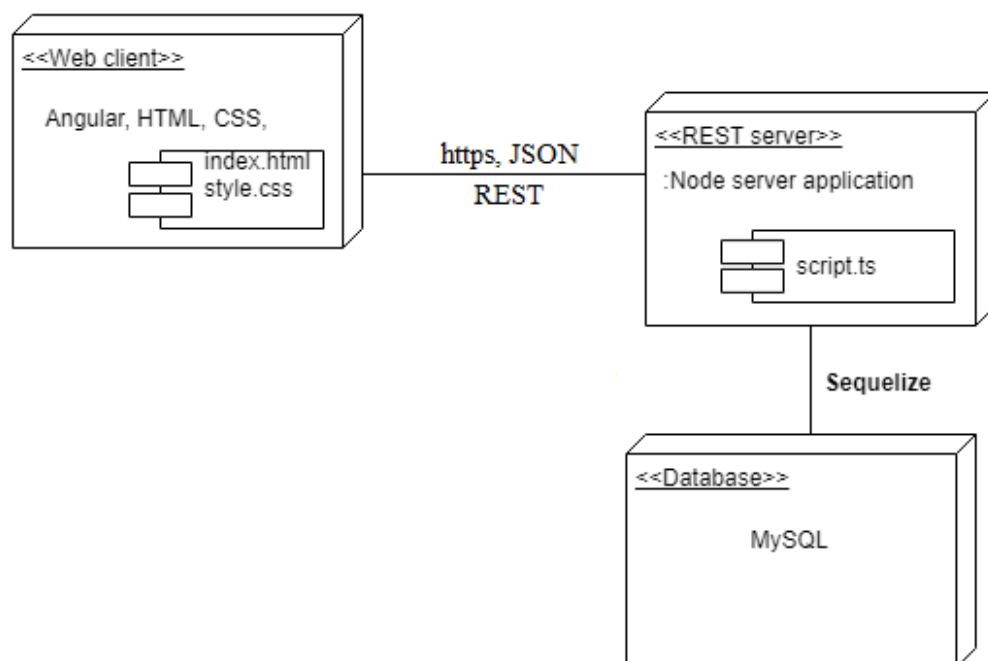


Рисунок 4.3 – Діаграма розгортання

Таким чином ми побачили, як виглядає графічне представлення програмної інфраструктури, на яку буде розгорнуто сайт та сервер.

На цьому закінчимо оглядовий екскурс по діаграмах зокрема і проектування в загалі. Варто зазначити, що процес проектування вже давно став стандартом розробки ПО, але часто доводиться стикатися з чудово написаною програмою, яка через відсутність нормальної документації обростає непотрібним побічним функціоналом, милицями, стає громіздкою і втрачає колишню якість. Саме тому, для побудови даної програмної системи була побудована така кількість діаграм [5, ст.43].

4.2 Побудова ER-діаграми

Які б системи не використовувалися, ER-діаграма наочно і точно відображає уявлення автора про дані [3, с.24]. Тому вона є гарним джерелом інформації для майбутнього процесу проектування логічної моделі даних.

Для проектування бази даних була обрана MySQL тому, що вона має ряд переваг: масштабування (досягається завдяки передовим алгоритмам використання дискового простору та удосконаленому процесорі запитів), досконала продуктивність, простота використання, направленість на веб. Якби системи не використовувалися, ER-діаграма наочно і точно відображає уявлення автора про дані. Для зберігання даних використовується реляційна база даних. Реляційна модель даних відображає інформацію у найбільш простій та зручній формі, забезпечує повну незалежність даних та базується на розвиненому математичному апараті, який дозволяє лаконічно описати основні операції над даними [4, ст.45].

В результаті проектування бази даних була отримана схема бази даних (див. рис. 4.4), що складається з п'яти сутностей.

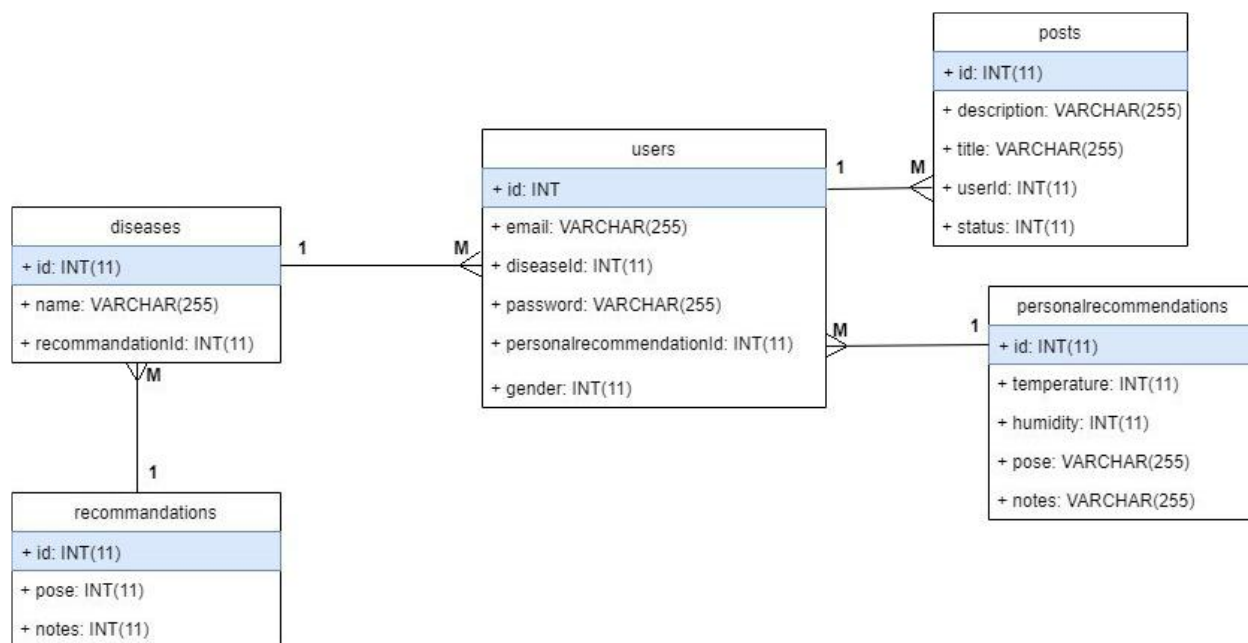


Рисунок 4.4 – ER-діаграма

У даній ER-діаграмі представлені такі сутності та їх атрибути, як:

а) користувач (ідентифікаційний номер користувача, електронна пошта, ідентифікаційний номер хвороби, пароль, ідентифікаційний номер персональної рекомендації, стаття);

- б) рекомендація (ідентифікаційний номер рекомендації, поза, нотатки);
- в) персональна рекомендація (ідентифікаційний номер персональної рекомендації, температура, вологість, поза, нотатки);
- г) хвороба (ідентифікаційний номер хвороби, назва, ідентифікаційний номер рекомендації);
- д) пост (ідентифікаційний номер посту, опис, заголовок, ідентифікаційний номер користувача, статус).

Одна з основних вимог до організації бази даних – це забезпечення можливості відшукування одних сутностей за значеннями інших, для чого необхідно встановити між ними певні зв'язки. Тому в діаграмі наглядно зображені наступні зв'язки між сутностями:

- а) один до багатьох між користувачем та персональною рекомендацією, тому що користувач може мати тільки одну рекомендацію, а рекомендація в свою чергу може бути використана багатьма користувачами;
- б) один до багатьох між користувачем та постами, тому що один користувач може розмістити скільки завгодно постів;
- в) один до багатьох між користувачем та хворобою, одна хвороба може бути у декількох людей;
- г) один до багатьох між хворобою та рекомендацією, тому що в рекомендації може поширюватися на багато хвороб, але для одної хвороби може бути тільки одна рекомендація.

Схема не потребує додаткових поліпшень, так як відношення знаходиться у другій нормальній формі, кожне поле містить у собі атомарне значення та між неключовими атрибутами немає транзитивних залежностей, тобто відношення знаходиться у третій нормальній формі.

Отже, в результаті проектування бази даних була отримана схема бази даних, що складається з чотирьох сутностей, що пов'язані між собою відношеннями типу один до багатьох та один до одного.

4.3 Архітектура програмної системи

Щодо архітектури – в даному проекті використовується багаторівнева архітектура проекту, тобто класична трирівнева система (див. рис. 4.5), що складається з наступних рівнів:

а) API layer (рівень взаємодії з веб сервером): це той рівень, з яким взаємодіє клієнт, посилаючи запити на веб сервер;

б) Business layer (рівень бізнес-логіки): містить набір компонентів, які відповідають за обробку отриманих від рівня уявлень даних;

в) Data Access layer (рівень доступу до даних): зберігає моделі, що описують використовувані суті.

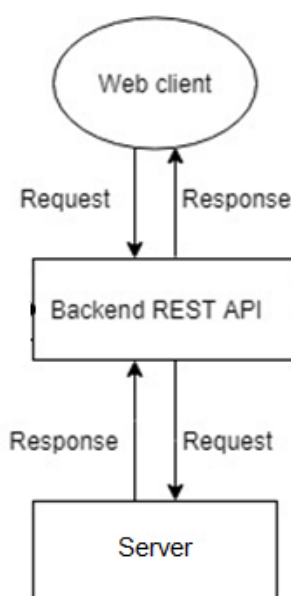


Рисунок 4.5 – Архітектура програмної системи

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних застосунків і передбачає взаємодію та обмін даними між ними.

Одна з найбільш важливих особливостей цього шаблону полягає в тому, що він робить можливим поділ відповідальності. Нам потрібно, щоб компоненти в додатках були максимально незалежними один від одного і мали лише таку кількість взаємозалежностей, яка піддається управлінню. В ідеальній ситуації кожен компонент нічого не знає про інші компоненти і взаємодіє з іншими областями програми лише за допомогою абстрактних інтерфейсів. Така взаємозалежність називається слабким зв'язком, і вона спрощує тестування і модифікацію додатків.

На рисунку 4.6 наведено діаграму діяльності (Activity diagram).

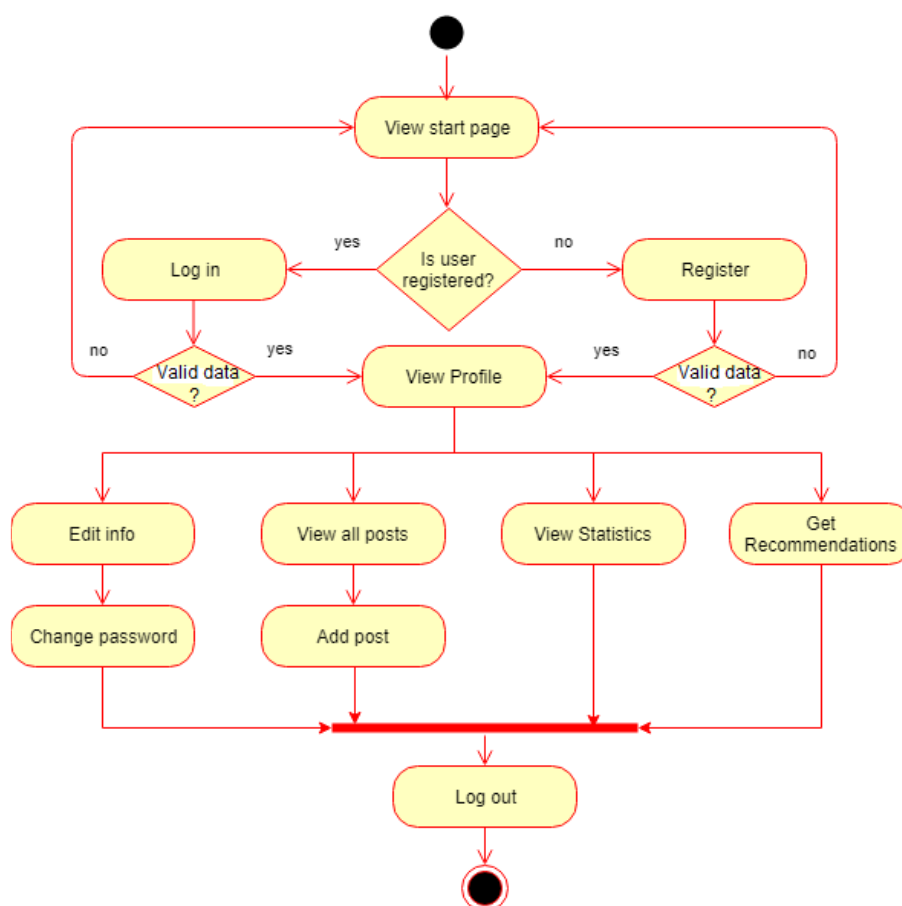


Рисунок 4.6 – Діаграма діяльності користувача

На даній діаграмі зображено можливі варіанти діяльності користувача на найвищому рівні абстракції.

4.4 Кодування програмного продукту

Серверна частина системи створювалася на платформі Node.js. Node.js дійсно хороший для створення швидких масштабованих мережеских додатків, оскільки дозволяє одночасно обробляти величезну кількість запитів з великою пропускнуою здатністю. Була створена база даних за допомогою системи MySQL, а також були написані контролери. Код було написано мовою програмування TypeScript в середовищі розробки Visual Studio Code. Для доступу до даних використовувався Sequelize [1, ст.29].

В даному проекті використовується багаторівнева архітектура проекту – класична трирівнева система, що складається з наступних рівнів: API layer (рівень взаємодії з веб сервером), Business layer (рівень бізнес-логіки), Data Access layer (рівень доступу до даних).

Також використовується механізм впровадження залежностей для більш гнучкої поведінки. Впровадження залежностей (Dependency Injection, DI) – чудова методика для реалізації слабо пов'язаних застосунків [2, ст.23]. Вона надає можливості для спрощення коду, вилучення та обробки залежностей між об'єктами і автоматичного створення екземплярів залежного об'єкта. Впровадження залежностей описує процес розробки застосунків – замість вказівки конкретних залежностей в застосунку під час розробки та створення необхідних об'єктів в коді під час виконання застосунок вирішує, які об'єкти йому потрібні, а потім створює і впроваджує їх в застосунок.

Нижче наведено код контролера, який обробляє запити і передає вхідні данні користувачу сервісу, отримуючи від нього нову порцію даних і відправляючи їх на клієнт.

```
export default class UserController {  
  public static async getAllUsers(req: Request, res:  
Response): Promise<void> {  
    const users: IUser[] = await UserService.getAllUsers();
```



```

        res.status(200).send(users);
    }

```

Далі наведено код додавання, видалення користувача та зміни пароля.

```

    public static async addUser(req: Request, res: Response):
    Promise<void> {
        const model = req.body;

        const user = await UserService.addUser(model);

        res.status(200).send(user);
    }

```

```

    public static async deleteUser(req: Request, res:
    Response): Promise<void> {
        const userId = req.params.id;

        await UserService.deleteUser(userId);

        res.status(200).send();
    }

```

```

    public static async changePassword(req: Request, res:
    Response) {
        const token = req.headers.authorization;
        const user: IUser = await
    UserService.getUserByToken(token);

        const newPass = req.body.newPass;
        const oldPass = req.body.oldPass;

        if (await compare(oldPass, user.password)) {
            const pass = await hash(newPass, 10);
            await UserService.changePassword(user.id, pass);

            res.status(200).send();
        } else {
            res.sendStatus(400);
        }
    }

```

Перед тим, як передати запит в контролер, відбувається перевірка на роль користувача, то, що він авторизувався, а також те, що на вхід надійшли валідні дані [3, ст.31].

Паролі хешуються за допомогою алгоритму BCrypt, який представлено у коді вище. Bcrypt – адаптивна криптографічна функція формування ключа, що використовується для безпечного зберігання паролів. Після хешування дані заносяться до бази даних. Доступ до певних функції мають лише користувачі з відповідними дозволенними ролями.

Для розробки веб-сайту було обрано мову програмування TypeScript, а також такі технології, як Angular, CSS та HTML.

Перше, що бачить перед собою користувач, коли зайшов на сайт – це форма для авторизації. Якщо не зареєстрований можна перейти до форми з реєстрацією, натиснувши на кнопку «Sign Up». Після цього користувач перенаправляється до сторінки з особистою інформацією (див. рис. 4.7).

The screenshot shows a web application interface for user management. The top navigation bar is blue with the 'HDreamer' logo and language selection buttons for 'En' and 'Ua'. A sidebar on the left lists navigation options: Profile, Users, Posts, New Posts, All Posts, and Statistic. The main content area displays the 'Profile' form, which includes input fields for 'First name*' (filled with 'Dima'), 'Last name*' (filled with 'Mylokhin'), and 'Email*' (filled with 'D@gmail.com'). There are also dropdown menus for 'Gender' (selected 'Male') and 'Disease' (selected 'Bruxism'), and a text input for 'Age' (filled with '19'). At the bottom of the form are three buttons: 'Change Password', 'Save', and 'Cancel'.

Рисунок 4.7 – Сторінка з особистою інформацією користувача

Користувач же в свою чергу має можливість отримати персональні рекомендації (див.рис.4.8).

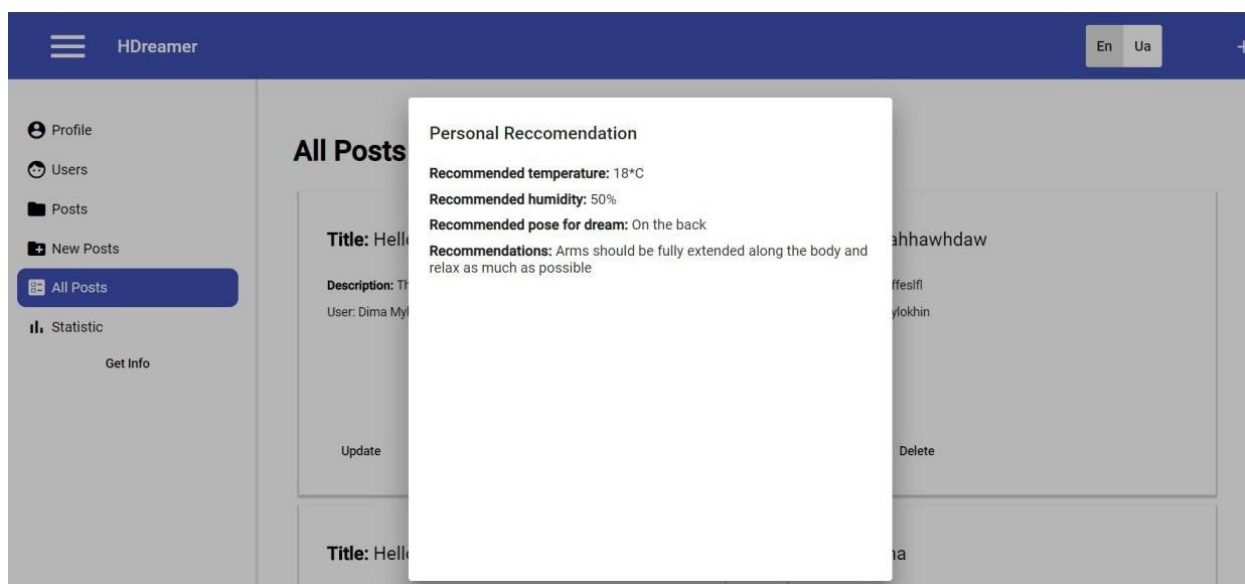


Рисунок 4.8 – Приклад виводу персональних рекомендацій

Користувач може редагувати особисті дані, а також змінювати пароль. Далі можна перейти на вкладку з усіма постами (див. рис. 4.9).

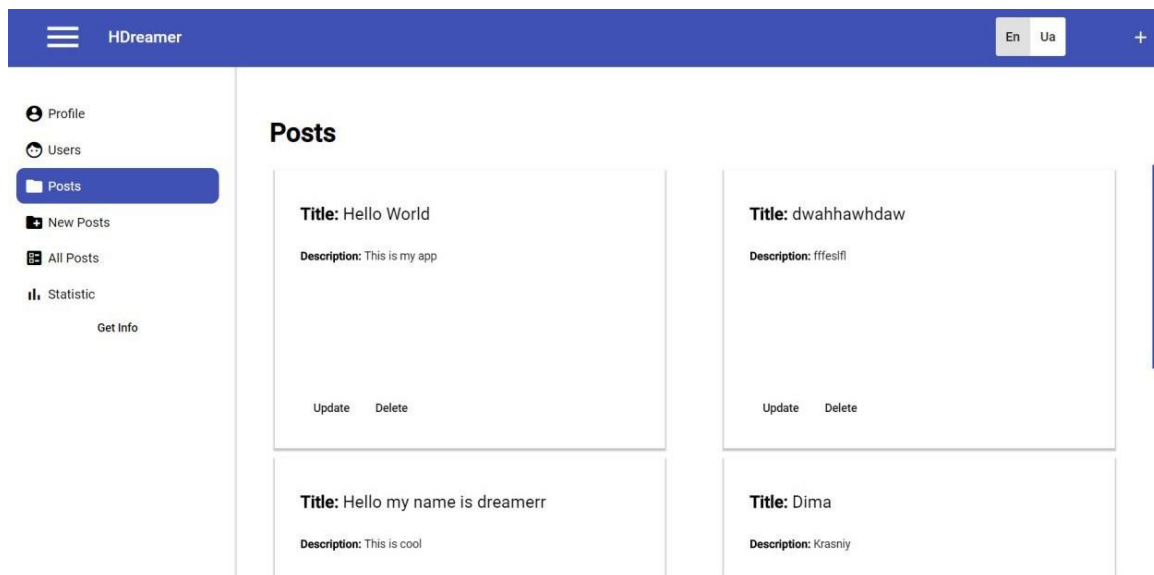


Рисунок 4.9 – Сторінка з усіма постами

Також є можливість додати новий пост, який буде додано до сайту тільки після того, як його підтвердить адміністратор. Для адміністратора нові пости відображаються на окремій сторінці «New Posts». Також адміністратору доступна

сторінка з усіма користувачами, на якій він може видалити одного з них або отримати його персональні рекомендації. Рекомендації включають в себе бажані показники температури та вологості, позу для сну, а також додаткові рекомендації, які залежать від пози. Рекомендації базуються на таких даних користувача, як рік, стать та хвороба. Крім того користувач може змінювати локалізацію сайту (українська та англійська мови), використовуючи перемикач у правому верхньому куточку екрану.

На сторінці «Statistic» можна переглянути два види діаграм. Одна базується на статистиці найбільш зустрічаємих хвороб, друга показує, які хвороби та у якому віці зустрічаються найчастіше (див. рис. 4.10).

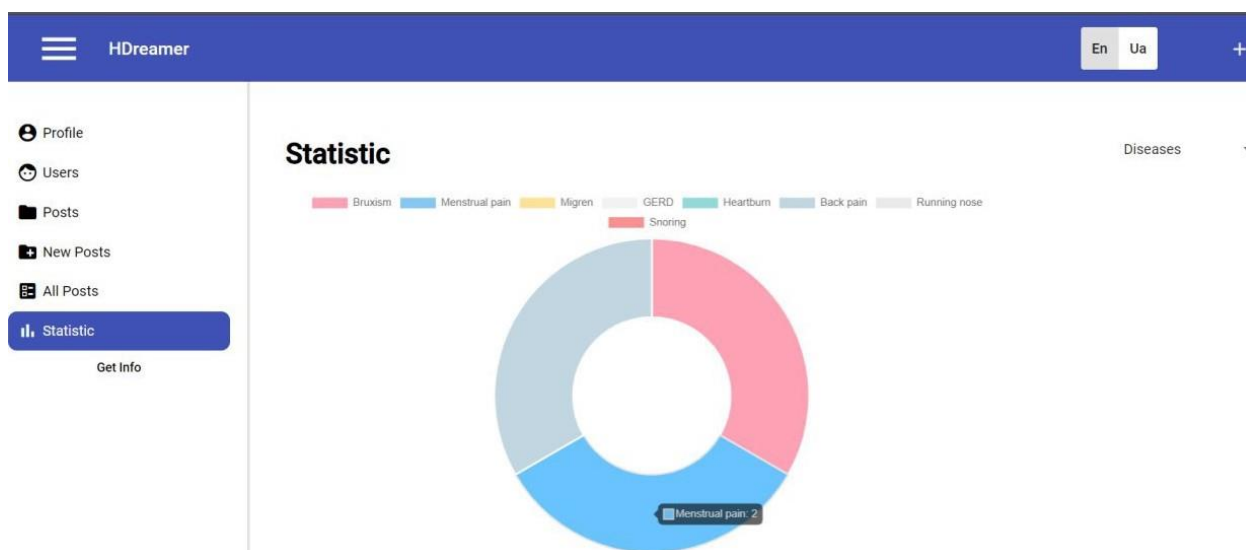


Рисунок 4.10 – Статистика найбільш зустрічаємих хвороб

Так як кожна частина пов'язана саме з одним й тим же параметром – хвороба, було використано кругову діаграму. Переключатися між діаграмами можна за допомогою випадаючого меню, назва якого змінюється в залежності від діаграми. Таким чином front-end частина проекту виконує усі потрібні звичайному користувачу функції та адміністратору функції для керування користувачами та постами.

ВИСНОВКИ

У результаті роботи була розроблена «Програмна система для автоматизації процесу підготовки до сну «HealthyDream». Дана програмна система буде допомагати користувачу контролювати якість свого сну, за допомогою корисних рекомендацій щодо вологості та температури, які повинні бути в кімнаті під час сну, а також рекомендації щодо того, у якій позі корисніше спати, в залежності від того, яка в людини історія хвороби.

На початку роботи було проведено аналіз предметної області, а також аналіз систем-аналогів, які вже присутні на ринку, виявлено їх основні переваги та недоліки.

У процесі концептуального моделювання предметної області та проектування були отримані UML-діаграми, що наочно описують поведінку системи. Також була визначена архітектура системи, технології, що використовувалися при розробці програмного продукту, отримана схема бази даних.

Програмна система має дві складові частини: сервер та веб-сайт.

Розробка велась з використанням мов програмування TypeScript для серверної частини програмної системи. Сайт було розроблено з використанням HTML5, CSS3 та мови Angular. Веб сайт відправляє запити на сервер по протоколу HTTPS у форматі JSON та приймає відповіді також у форматі JSON. В якості бази даних був обраний сервіс MySQL.

Розроблена програмна система є лише прототипом та не може в поточному стані використовуватися в реальних умовах. Зараз вона дозволяє виконувати базові функції такі, як авторизація, реєстрація, перегляд персональної інформації користувача, створення нових постів, перегляд рекомендацій. Незважаючи на це, розвиток системи є досить перспективним, тому в наш час прискореного ритму життя, здоровий сон є дуже важливим.

ПЕРЕЛІК ПОСИЛАНЬ

1. Sente Lab [Електронний ресурс] – Режим доступу до ресурсу: <https://www.sente-lab.com/>. (дата звернення: 10.12.2018)
2. Use Case – Wikipedia [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Use_case (дата звернення: 15.11.2018)
3. Боггс М. UML и Rational Rose / М. Боггс. – Москва: РГГУ, 2016. – 438 с.
4. Бен-Ган И. Microsoft SQL Server 2012. Основы T-SQL / Ицик Бен-Ган. – Москва: Эксмо, 2015. – 400 с.
5. Вигерс Карл Разработка требований к программному обеспечению / Пер. с англ. — М.: Издательско-торговый дом «Русская Редакция», 2004. — 576с.
6. Фаулер М. UML. Основы, 3-е издание. — Символ-плюс, 2015. — 237с