МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри, канд. техн. наук,
доцент _____ Є. О. Давиденко
«___»____2022 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

ГЕЙМИФІКОВАНИЙ ТРЕКЕР ЗВИЧОК

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ.1 – 409.21810904

Студентка _____ С. В. Бондаренко
«___»_____2022 р.

Керівник завідувач кафедри ІПЗ, канд. техн. наук, доцент _____ Є. О. Давиденко
«___»_____2022 р.

Консультант канд. техн. наук, доцент _____ А. О. Алексєєва
«___»_____2022 р.

Миколаїв 2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ЗАТВЕРДЖУЮ
Зав. кафедри
_____ Є. О. Давиденко
« __ » _____ 2022 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи бакалавра

Видано студентці групи 409 факультету комп'ютерних наук

Бондаренко Стефанії Віталіївні .
(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи:

Геймифікований трекер звичок

Затверджена наказом по ЧНУ ім. П.Могили від « 01 » грудня 2021 р. № 314

2. Строк представлення кваліфікаційної роботи: « __ » _____ 202__ р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Очікуваним результатом є геймифікований трекер звичок.

4. Перелік питань, що підлягають розробці:

- дослідження предметної області та аналіз існуючих аналогів;
- формування специфікації вимог до програмного забезпечення;
- визначення архітектури для проєктування програмного забезпечення;
- моделювання та проєктування програмного забезпечення;
- розробка програмного забезпечення;
- здійснення тестування роботи програмного забезпечення;
- проведення аналізу результатів розробки;

5. Перелік графічних матеріалів:

презентація.

6. Завдання до спеціальної частини

Аналіз охорони праці на робочих місцях фахівців з інформаційних технологій.

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
Алєксєєва А.О.	Кафедра екології	Спеціальна частина з охорони праці

Керівник роботи канд. техн. наук, доцент Давиденко Євген Олександрович
(посада, прізвище, ім'я, по батькові)

_____
(підпис)

Завдання прийнято до виконання

Бондаренко Стефанія Віталіївна
(прізвище, ім'я, по батькові студента)

_____
(підпис)

Дата видачі завдання « ___ » _____ 202__ р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

Тема кваліфікаційної роботи:

Геймифікований трекер звичок

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КРБ	03.10.2021 р.	04.10.2021 р	виконано
2.	Огляд літератури за темою роботи	06.10.2021 р.	08.10.2021 р.	виконано
3.	Складання календарного плану КРБ	09.10.2021 р.	10.10.2021 р.	
4.	Аналіз предметної області	15.10.2021 р.	15.10.2021 р.	виконано
5.	Розробка проєктних рішень	03.11.2021 р.	04.11.2021 р.	виконано
6.	Моделювання та конструювання ПЗ	05.11.2021 р.	10.11.2021 р.	виконано
7.	Кодування, тестування розробленого ПЗ, аналіз результатів тестування, розробка керівництва користувача	16.11.2021 р.	22.02.2022 р.	виконано
8.	Розробка спеціальної частини з охорони праці	14.05.2022 р.	28.05.2022 р.	виконано
9.	Оформлення КРБ та презентації	07.06.2022 р.	09.06.2022 р.	виконано
10.	Відгук керівника КРБ	10.06.2022 р.	11.06.2022 р.	виконано
11.	Попередній захист	13.05.2022 р.	15.06.2022 р.	виконано
12.	Завершення оформлення КРБ та презентації	15.06.2022 р.	17.06.2022 р.	виконано
13.	Рецензування	18.06.2022 р.	19.06.2022 р.	виконано
14.	Захист кваліфікаційної роботи	27.06.2022 р.	27.06.2022 р.	виконано

Розробила студентка Бондаренко Стефанія Віталіївна _____
(прізвище, ім'я, по батькові) *(підпис)*

« ___ » _____ 2022 р.

Керівник роботи канд. техн. наук, доцент Давиденко Є. О. _____
(посада, прізвище, ім'я, по батькові) *(підпис)*

« ___ » _____ 2022 р.

АНОТАЦІЯ

до кваліфікаційної роботи бакалавра

«Геймифікований трекер звичок»

Студентка 409 гр.: Бондаренко Стефанія Віталіївна

Керівник: канд. техн. наук, доцент Давиденко Є. О.

Дана робота присвячена розробці програмного забезпечення для автоматизації процесу формування звичок з підвищенням власної ефективності та самомотивації, ґрунтуючись на роботі з базами даних: коучів, методичних рекомендацій та досягнень.

Об’єкт роботи: процес формулювання корисних звичок з підвищенням власної ефективності та самомотивації.

Предмет роботи: програмні засоби створення геймифікованого трекеру звичок.

Мета: автоматизація процесу формування звичок за рахунок розробки програмного забезпечення геймифікованого трекеру звичок.

Кваліфікаційна робота бакалавра складається з вступу, чотирьох розділів, висновків та переліку джерел посилання.

У вступі визначається актуальність теми, що приймається за мету та невеликий огляд поставленої задачі, предмет дослідження та об’єкт роботи.

У першому розділі описується аналітична частина, тобто огляд існуючих застосунків-аналогів трекерів звичок, визначення функціоналу, переваг та недоліків програмного забезпечення, обґрунтовується план виконання завдання. Наступною частиною розділу є формування та опис специфікації вимог до програмного забезпечення, що розробляється.

У другому розділі описується процес розробки проєктних рішень, що забезпечують виконання пунктів специфікації вимог до програмного забезпечення, тобто моделювання об’єкту та предмету дослідження, а також функціональні та інформаційні моделі програмного забезпечення. В результаті процесу розробки складено детальний алгоритм вирішення поставленої задачі.

У третьому розділі описується результат виконаної роботи з

конструювання та моделювання програмного забезпечення, включаючи вибір технологій, мови програмування, та компонентів застосунку, тобто бібліотек, плагінів та ін, розробку UML-діаграм та опис інтерфейсів геймифікованого трекеру звичок.

У четвертому розділі демонструється проведена робота з кодування та тестування застосунку геймифікованого трекеру звичок, крім того описується аналіз отриманих результатів під час тестування програмного забезпечення.

У висновках проводиться аналіз роботи та отриманих результатів.

Кваліфікаційна робота бакалавра викладена на 59 сторінок, вона містить 4 розділи, 36 ілюстрацій, 18 таблиць, 23 джерел в переліку посилань.

Ключові слова: *геймифікований трекер, формування звичок, створення вебзастосунку, розробка на Laravel, робота з коучем, підвищення ефективності та самомотивації, автоматизація процесу формування звичок, розробка програмного забезпечення.*

ABSTRACT

of the Bachelor's Thesis

"Gamified habits tracker"

Student: Bondarenko Stefaniia

Supervisor: Candidate of Technical Sciences (Ph. D.), Associate Professor

Davydenko Y. O.

This work is devoted to the development of software to automate the process of forming habits with increasing efficiency and self-motivation, based on working with databases: coaches, guidelines, and achievements.

The object of work: the process of formulating useful habits with increasing efficiency and self-motivation.

The subject of work: software tools for creating a gamified habit tracker.

Objective: to automate the process of habit formation through the development of gamified habit tracker software.

The qualification work of the bachelor consists of an introduction, four chapters, conclusions, and list of sources links.

The introduction determines the relevance of the topic to be taken as a goal and a brief overview of the task, the subject of research, and the object of work.

The first section describes the analytical part, ie an overview of existing applications-analogs of habit trackers, determining the functionality, advantages, and disadvantages of software, and substantiating the plan for the task. The next part of the section is the formation and description of the specification of the requirements for the software being developed.

The second section describes the process of developing design solutions that ensure compliance with the specification of software requirements, ie modeling of the object and subject of research, as well as functional and informational models of software. As a result of the development process, a detailed algorithm for solving the problem was developed.

The third section describes the results of the software design and modeling work, including the choice of technologies, programming languages, application components, ie libraries, plugins, etc., development of UML diagrams, and description of gamified habit tracker interfaces.

The fourth section demonstrates the work done on coding and testing the application of gamified habit tracker, in addition, describes the analysis of the results obtained during software testing.

The conclusions analyze the work and the results obtained.

The qualification work of the bachelor is presented on 59 pages, it contains 4 sections, 36 illustrations, 18 tables, 23 sources in the list of references.

Keywords: *gamified tracker, habit formation, web application creation, development on Laravel, work with a coach, increase of efficiency and self-motivation, automation of habit formation process, software development.*

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ 4
ВСТУП 5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 7
1.1 Огляд застосунків-аналогів геймифікованого трекеру звичок 7
1.2 Аналіз системи, що розробляється 12
1.3 Специфікація вимог до програмного забезпечення «HabiTracker» 14
Висновки до розділу 1 18
2 МОДЕЛЮВАННЯ ГЕЙМИФІКОВАНОГО ТРЕКЕРУ ЗВИЧОК 19
2.1 Етапи реалізації проєкту 19
2.2 Створення Use case 20
2.3 Алгоритм роботи програмного забезпечення 24
2.4 Розробка діаграми розгортання 27
2.5 Побудова та використання діаграм взаємодії 28
2.6 Розробка шаблону MVC для керованого інтерфейсу 30
Висновки до розділу 2 32
3 ПРОЄКТУВАННЯ ВЕБЗАСТОСУНКУ ТА ОГЛЯД СТЕКУ ЗАДІЯНИХ ТЕХНОЛОГІЙ 33
3.1 Розробка UML-діаграм 33
3.1.1 Діаграма класів 34
3.1.2 Діаграми станів та переходів 36
3.1.3 Діаграми компонентів 37
3.1.4 Діаграма пакетів 39
3.2 Оглядання технологій 40
3.2.1 Мови програмування 41
3.2.2 Технології front-end розробки 42
3.2.3 Технології back-end розробки 43
Висновки до розділу 3 45

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ ТА ПРОВЕДЕННЯ ТЕСТУВАННЯ 46
4.1 Огляд дизайну вебзастосунку 46
4.2 Реалізація програмних компонентів бізнес-логіки геймифікованого трекеру звичок 49
4.2.1 Реєстрація, авторизація та адаптивний інтерфейс 49
4.2.2 Головна сторінка (Home page) 51
4.2.2 Додавання звичок і використання медіа для медитації 53
4.3 Тестування вебзастосунку 55
Висновки до розділу 4 59
ВИСНОВКИ 60
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 61

ПЕРЕЛІК СКОРОЧЕНЬ

БД	–	база даних
КРБ	–	кваліфікаційна робота бакалавра
ОС	–	операційна система
ПЗ	–	програмне забезпечення
ПК	–	персональний комп’ютер
СКБД	–	система керування базами даних
ЦА	–	цільова авдиторія
API	–	application programming interface
CRUD	–	create, read, update, delete
CSS	–	cascading style sheets
HTML	–	hypertext markup language
JS	–	JavaScript
MIT	–	Massachusetts Institute of Technology
MVC	–	model-view-controller
PHP	–	hypertext preprocessor
SQL	–	structured query language
QR code	–	quick response code
UI	–	user interface
UML	–	unified modeling language
UX	–	user experience
Web app	–	web application
WAP	–	wireless application protocol

ВСТУП

Актуальність теми кваліфікаційної роботи бакалавра зумовлена сучасними тенденціями особистісного самовдосконалення. З народження людина розвивається та вдосконалюється, спочатку в цьому допомагають батьки, пізніше – вчителі, викладачі та тренери, але кожна людина прагне досягти чогось більшого, ніж отримати базові освітні знання, оскільки це сприяє її розвитку як особистості та допомагає виділятися від інших. Для того, щоб чогось досягти треба діяти, але тайм-менеджмент хоч і дуже важливий, але не кожен має змогу керувати власним часом вміло. Для отримання результату важливо займатися справою регулярно та приділяти достатню кількість часу.

Методика набуття звички хоч і проста, але в сучасному житті є проблема розподілу часу та незручності ведення графіку справ. Саме для цього використовуються трекери звичок, вони допомагають керувати своїм часом з користю, оскільки в них можна додавати звички та керувати ними, відмічати прогрес та відслідковуючи регулярність виконання поставлених цілей, а головне – це зручно та мобільно, також додатково можна скористуватися допомогою коуча та отримати досягнення в застосунку, що дуже допомагає та мотивує.

Об'єкт роботи: процес формування корисних звичок з підвищенням власної ефективності та самомотивації.

Предмет роботи: програмні засоби створення геймифікованого трекеру звичок.

Мета: автоматизація процесу формування звичок за рахунок розробки програмного забезпечення геймифікованого трекеру звичок.

Відповідно для досягнення визначеної мети необхідно вирішити наступні **завдання**:

- аналіз застосунків-аналогів;
- відокремлення проблем та пошук їх вирішення;
- визначення функціоналу застосунку;

– створення блок-схем алгоритму роботи геймифікованого трекеру звичок та UML-діаграм;

– розробка дизайну досягнень та макету застосунку;

– збір та аналіз інформації методичних рекомендацій;

– розробка front-end частини вебзастосунку на базі технології Vue.js, Bootstrap;

– розробка back-end частини вебзастосунку на базі технології Laravel, MySQL.

Сфера застосування: геймифікований трекер звичок можна використовувати як у повсякденному житті, так і в сфері спорту, навчання, хобі та творчості, здорового способу життя, зокрема налагодження сну, медитації, харчування, догляд за шкірою тощо. Застосунок також корисний для коучів, оскільки є гарною допомогою при пошуку клієнтів та при зборі відгуків й формуванні рейтингу тренера.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд застосунків-аналогів геймифікованого трекеру звичок

Аналіз застосунків-аналогів до ПЗ, що розробляється, є важливою частиною, оскільки це допомагає виявити основний затребуваний функціонал та недоліки застосунків, що є конкурентами, для їх усунення при розробці та підвищення якості власного ПЗ. Конкурентоспроможний застосунок має не лише містити в собі подібний функціонал інших застосунків, але й доповнювати його для охоплення більшої кількості ЦА або покращення його зручності користування. Для аналізу аналогів обрано застосунки: Balance (табл. 1.1), Habitbull (табл. 1.2), Habitshare (табл. 1.3).

Balance

Застосунок створений для розвитку практик медитацій. Крім того «Balance» має зрозумілий та лаконічний інтерфейс (рис. 1.1), але основним недоліком застосунку є вузькоспеціалізований функціонал, оскільки можна набувати лише тих звичок, що пов’язані з медитативними практиками. Практики медитацій розкладені на конкретні навички, щоб їх можна було тренувати окремо [1]:

- фокусування дихання;
- концентрація;
- візуалізація;
- розслаблення;
- налагодження сну тощо.

Таблиця 1.1 – Опис 3tier application «Balance»

Назва	**Balance**
Архітектура	3tier application
Виробник	Приватна компанія *Elevate Labs*
Мова реалізації	Java

Кінець таблиці 1.1

Функції	1. Відповідати на питання щодо особистого досвіду, медітації, цілей, проблем. 2. Авдіотека з більш ніж 1000 файлів для медитації. 3. Застосунок персоналізується і підбирає більш ефективні рекомендації для кожного користувача. 4. Бібліотека матеріалів для медитації (збірки практик, порад, музики, ігри). 5. Інтерактивний сингл Wind Dow. 6. Імерсийні медитації.
Переваги	1. Безкоштовний персоналізований застосунок, доступний для Android, iOS. 2. Є рекомендації від тренерів по медитації, є можливість слідкувати за прогресом медитативних навичок.
Недоліки	1. Застосунок спрямований лише на формування звичок, що стосуються медитативних практик, є платний функціонал. 2. Великий проміжок часу для завантаження файлів. 3. Відсутність можливості роботи з коучем.
Вебсайт	www.balanceapp.com

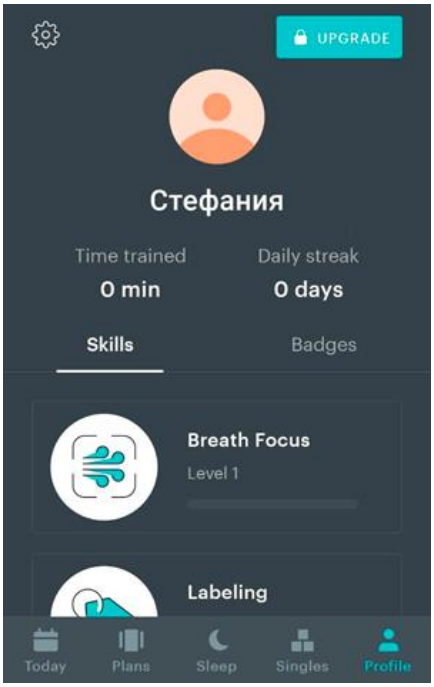


Рисунок 1.1 – Вигляд інтерфейсу застосунку «Balance»

Habitbull

Застосунок призначений для набуття звичок в різних категоріях:

- мистецтво;
- здоров’я та фітнес;
- домогосподарство та тварини;
- фінанси;
- самовдосконалення;
- робота та навчання тощо.

Таблиця 1.2 – Опис 3tier application «HabitBull»

Назва	**HabitBull**
Архітектура	3tier application
Виробник	Приватна компанія *App Holdings*
Мова реалізації	Java
Функції	1. Спілкування на спеціалізованих форумах. 2. Можливість увімкнути нагадування щодо нових звичок. 3. Експорт своїх даних у форматі CSV. 4. Віжет для екрана (Лише на Android). 5. Створення будь-яких цілей. 6. Перевірка та вивчення особистих даних за допомогою інстументів глибокої аналітики. 7. Синхронізація декількох пристроїв.
Переваги	1. Застосунок безкоштовний як для Andriod, так і для iOS. 2. Обширний функціонал, окремий календар для кожної звички, мотиваційні зображення, відсутність реклами. 3. Можливість експорту даних.
Недоліки	1. Застарілий інтерфейс, інколи складно знайти необхідний функціонал, оскільки вони згруповані не завжди асоціативно зрозуміло, деякі функції приховано в меню. 2. Відсутність можливості роботи з коучем. 3. Наявність рекламних банерів.
Вебсайт	www.habitbull.com

Інтерфейс застосунку простий (рис. 1.3), але має лише вкладку із всіма звичками та вкладки для кожної звички окремо. Перелік тем на форумі знаходиться у загальній вкладці, без можливості пошуку по форумам, що ускладнює роботу користувача із застосунком.

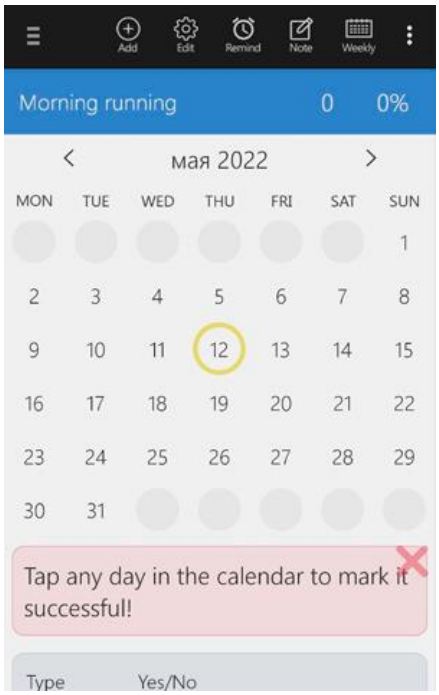


Рисунок 1.2 – Вигляд інтерфейсу застосунку «HabitBull»

Habitshare

Застосунок призначений для набуття звичок (рис. 1.3), на відміну від «Habitbull», не містить певних категорій, проте особливістю застосунку є можливість додавати користувачів у список друзів та ділитися з ними інформацією про звички [2].

Таблиця 1.3 – Опис 3tier application «HabitShare»

Назва	**HabitShare**
Архітектура	3tier web application
Виробник	Приватна компанія Luke Bickston
Мова реалізації	Java
Функції	1. Запланувати нагадування, щоб пам'ятати про нові звички. 2. Створення власних навичок, що відповідають особистим конкретним цілям, включаючи щоденні та щотижневі звички.

Кінець таблиці 1.3

Функції	3. Додавання друзів для додаткової відповідальності та мотивації. 4. Експорт своїх даних у форматі CSV.
Переваги	1. Застосунок безкоштовний для Android та iOS. 2. Інтуітивно зрозумілий дизайн. 3. Можливість експорту даних.
Недоліки	1. Відсутність мотиваторів та теоретичної інформації для користувачів. 2. Не можна синхронізувати декілька пристроїв. 3. Відсутність можливості роботи з коучем.
Вебсайт	habitshareapp.com

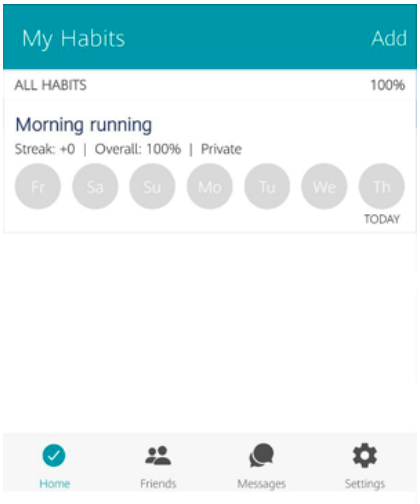


Рисунок 1.3 – Вигляд інтерфейсу застосунку «HabitShare»

Додавати друзів можна надіславши посилання на пошту, за номером телефону або скориставшись QR-кодом. Є можливість експорту даних звичок. Інтерфейс застосунку достатньо мінімалістичний, містить 4 основні розділи:

– home (домашня сторінка);
– friends (друзі);
– messages (повідомлення);
– settings (налаштування).

1.2 Аналіз системи, що розробляється

Розробка геймифікованого трекеру звичок «HabiTracker» (рис. 1.4) призначена для формування корисних звичок. Користувач має потребу в набутті нової звички, додає її у застосунок з відповідними опціями, крім того користувач потребує допомоги коуча – звертається до нього, отримує відповідні поради та рекомендації. Також користувач має змогу самостійно шукати інформацію у вже підготованих постах, мотиваторах та бібліотеки матеріалів для медитації.

Система повинна забезпечувати багатокористувацький режим роботи. Інтерфейс має бути сумісним з ОС Windows (вище 10 версії) та із смартфонами з ОС iOS та Android (вище версії 7.1). Система повинна бути в працездатному стані 18 годин на добу, 7 днів на тиждень.

Таблиця 1.4 – Опис системи, що розробляється

Функції	1. реєстрація користувачів; 2. авторизація користувачів; 3. розміщення та пошук мотиваторів; 4. додавання цілей; 5. вибір коуча; 6. відмічання прогресу виконання цілей; 7. допомога з медитативними практиками (бібліотека матеріалів, таймери і т.д.); 8. перегляд та даних прогресу; 9. чат з коучем; 10. історія звичок; 11. календар з актуальними та завершеними звичками;
Користувачі	1. коуч; 2. адміністратор; 3. користувач.
Сценарії роботи	1. Користувач встановлює застосунок з метою набуття нової звички \ позбавлення від старої, реєструється. 2. Користувач додає необхідні цілі в застосунок.

Кінець таблиці 1.4

Сценарії роботи	3. Користувач обирає коуча відповідно із власними вимогами. 4. Коуч підбирає поради та корисну інформацію \ завдання для користувача, щоб допомогти в роботі із поставленими користувачем цілями. 5. Користувач використовує рекомендації коуча. 6. Користувач відмічає прогрес. 7. Користувач має труднощі – звертається до коуча. 8. Коуч допомагає із подоланням труднощів. 9. Адміністратор \ коуч додає потенційно цікаву інформацію для користувача в розділ з мотиваторами та рекомендаціями. 10. Користувач продовжує відмічати прогрес, закінчує процес набуття нової звички або позбавлення від старої непотрібної.

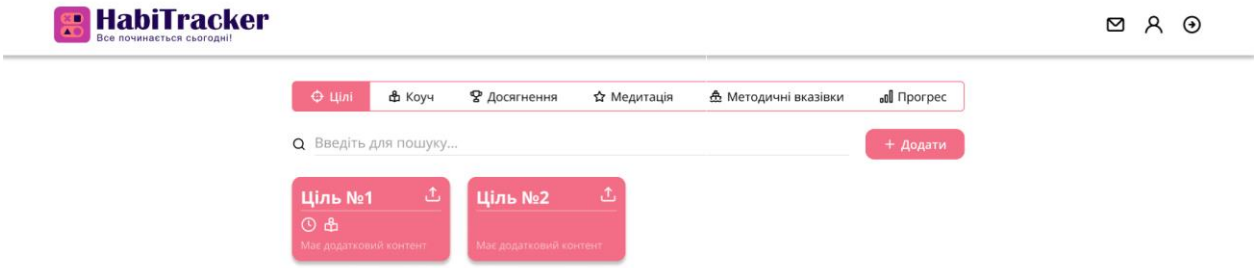


Рисунок 1.4 – Макет інтерфейсу застосунку «HabiTracker»

Для створення mock-up використано спеціалізований застосунок «Balsamiq Wireframes». Для користувача доступно 6 основних розділів:

– цілі – додавання, редагування та видалення цілей, відмічання прогресу;
– коуч – перегляд наданих порад;
– досягнення – перегляд отриманих досягнень;
– медитація – використання бібліотеки матеріалів для медитації;
– методичні вказівки – перегляд базових методичних вказівок;
– прогрес – перегляд прогресу щодо виконання поставлених цілей.

1.3 Специфікація вимог до програмного забезпечення «HabiTracker»

ПРИЗНАЧЕННЯ ТА МЕЖІ ПРОЄКТУ

Призначення системи (застосунку), для якої розробляється програмне забезпечення

Призначенням застосунку є автоматизація процесу формування звичок за рахунок розробки ПЗ геймифікованого трекеру звичок.

Погодження, що ухвалені в програмній документації

Було погоджено, що для створення загального ПЗ та його злагодженої роботи будуть використовуватися допоміжні фреймворки – Laravel.

Межі проєкту ПЗ

Крайня дата завершення роботи над ПЗ – 10.06.2022р.

ЗАГАЛЬНИЙ ОПИС

Сфера застосування

Дане ПЗ не має обмежень у сферах його застосування, так застосунок можна використовувати як у повсякденному житті, так і в сфері спорту, навчання, хобі та творчості, здорового способу життя, зокрема налагодження сну, медитації, харчування, догляд за шкірою тощо.

Характеристики користувачів

Основні характеристики користувачів: наявність смартфону, планшету або ПК та доступу до мережі Інтернет.

Загальна структура і склад системи

Основні частини для створення програмного забезпечення: сервер, БД, вебзастосунок, API.

Загальні обмеження

Обмеження для роботи з ПЗ – наявність доступу до мережі Інтернет.

ФУНКЦІЇ СИСТЕМИ ГЕЙМИФІКОВАНОГО ТРЕКЕРУ ЗВИЧОК

Функція додавання нової звички

Опис функції

Функція додавання звички допомагає користувачу слідкувати за процесом

набуття, отримувати поради від коуча стосовно відповідної звички.

Вхідна і вихідна інформація

Вхідна інформація – назва звички, детальний опис, дата завершення;

вихідна інформація – методичні рекомендації, поради коуча.

Функціональні вимоги

Бази даних з методичними рекомендаціями, порадами та доступ до мережі Інтернет.

Функція пошуку медитативних матеріалів з фільтрацією та сортуванням

Опис функції

Функція пошуку допомагає користувачу знайти матеріали з бібліотеки медитативних матеріалів за назвою, фільтрація зі свого боку допомагає відобразити матеріали відповідно до обраної категорії, сортування – за відповідним критерієм.

Вхідна і вихідна інформація

Вхідна інформація – назва, категорія (музика, відео);

вихідна інформація – медитативні матеріали, що задовольняють критерії пошуку.

Функціональні вимоги

База даних з матеріалами для медитації та доступ до мережі Інтернет.

Функція отримання досягнення

Опис функції

Функція отримання досягнення демонструє загальний прогрес користувача та є додатковим мотиватором, досягнення нараховуються системою автоматично.

Вхідна і вихідна інформація

Вхідна інформація – дані звичок користувача;

вихідна інформація – повідомлення про отримання досягнення з описом.

Функціональні вимоги

БД зі звичками, досягненнями користувача та доступ до мережі Інтернет.

ВИМОГИ ДО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

Джерела і зміст вхідної інформації (даних)

В даному ПЗ основним джерелом вхідної інформації є користувач. Який зі свого боку має задавати вручну необхідні дані, у випадку застосунку геймифікованого трекеру звичок – це дані звичок, відмічання прогресу та назва й категорія медитативних матеріалів.

Нормативно-довідкова інформація (класифікатори, довідники тощо)

Вимоги до даного пункту відсутні.

Вимоги до способів організації, збереження та ведення інформації

Обмін даними відбувається через RESTful API, БД – MySQL.

ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

Вимоги до технічного забезпечення не мають жорстких обмежень. Комп’ютер чи ноутбук з процесором, що підтримує SSE2, оскільки без підтримки даного набору інструкцій неможливо працювати з сучасними браузерами, включно з Google Chrome та Opera, а також з оперативною пам’яттю об’ємом не меньше 8 ГБ для підтримки необхідних програм для розробки.

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Архітектура програмної системи

Архітектура складається з клієнтської частини, серверної частини та БД.

Системне програмне забезпечення

Застосунок має бути побудованим з використанням фреймворку Laravel. Для написання front-end частини використано Bootstrap, Vue.js, для back-end частини – Laravel, MySQL. Проце обміну даними має відбуватися з використанням RESTful API. В якості БД для застосунку обрано MySQL.

Мережне програмне забезпечення

Для створення ПЗ використано ОС Windows 10, у якості HTML-редактору обрано Atom, для зручного перегляду вебсторінок – Opera та Google Chrome.

Програмне забезпечення ведення інформаційної бази

Через MySQL виконуються CRUD-операції з базою застосунку.

Мова і технологія розробки ПЗ

Програмне забезпечення має розроблюватися з використанням фреймворку Laravel. Мови розробки – PHP та JavaScript.

ВИМОГИ ДО ЗОВНІШНІХ ІНТЕРФЕЙСІВ

Інтерфейс користувача

Інтерфейс має задовольняти усі вимоги UX та UI дизайну, що дозволить користувачу затратити якомога менше часу на розуміння процесу роботи системи. Шаблон вебзастосунка повинен бути розділений на дві частини, де перша частина це пошукова панель, де розміщенні фільтри для пошуку, вона розміщена зліва, друга частина є основний контент який змінюється відповідно до заданих фільтрів.

Апаратний інтерфейс

Апаратним інтерфейсом є девайс користувача (ПК, смартфон чи планшет), що буде використовуватися для роботи з вебзастосунком.

Програмний інтерфейс

Laravel – фреймворк для вебзастосунків з достатньо виразним синтаксисом для розробки на мові програмування PHP. Вагомими перевагами фреймворку є наявність REST-контролерів для обробки запитів та Blade, який є движком для шаблонів, і підтримка видів тестування: Browser, Feature, Unit.

Комунікаційний протокол

Застосунок передбачає використання мережних протоколів WAP – протокол безпровідної передачі даних та TCP/IP.

ВЛАСТИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Доступність

ПЗ є доступним для будь-якого користувача, що має бажання використовувати вебзастосунок, за умови наявності у користувача апаратного інтерфейсу та доступу до мережі Інтернет.

Супроводжуваність

Вебзастосунок не потребує супроводжуваності.

Переносимість

Програмне забезпечення може працювати на ОС Windows (вище 10 версії) та із смартфонами з ОС iOS та Android (вище версії 7.1).

Продуктивність

Продуктивність роботи ПЗ залежить від швидкості підключення до мережі Інтернет. Час виконання запитів не має перевищувати терміну в 2 секунди.

Надійність

Дані надані користувачем при реєстрації, а також використанні застосунку повинні бути приватними. ПЗ має виключити можливості отримання персональних даних користувача будь-якими методами для інших користувачів. Крім того для отримання доступу до власних даних користувач має бути авторизованим у системі.

Безпека

Процедура автентифікації має відбуватися з процесом токен авторизації.

Висновки до розділу 1

В першому розділі кваліфікаційної роботи бакалавра оглянуто та проведено аналіз існуючих застосунків-аналогів геймифікованого трекеру звичок, завдяки чому розвинуто навички пошуку та усунення проблем застосунку, набуто практичних навичок з формування специфікації вимог до програмного забезпечення. Для проведення аналізу розглянуто функціонал, інтерфейс, переваги та недоліки застосунків-аналогів. Крім того, проведено аналіз системи, що розробляється, при якому визначено користувачів, базовий функціонал та сценарії роботи вебзастосунку.

Розділ містить специфікацію вимог до програмного забезпечення, що розробляється. В даній частині розділу представлено повний опис поведінки ПЗ. До опису поведінки включено множину як функціональних, так і нефункціональних вимог для опису взаємодії роботи користувачів з ПЗ.

2 МОДЕЛЮВАННЯ ГЕЙМИФІКОВАНОГО ТРЕКЕРУ ЗВИЧОК

2.1 Етапи реалізації проєкту

Розробка вебзастосунку полягає в створенні дизайну, front-end та back-end частин розробником або командою розробників та може бути реалізована як з з виділенням значного фінансування так і з обмеженим бюджетом. Тривалість розробки залежить від складності проєкту та складу розробників, а також – їх досвіду. Розробка вебзастосунку зазвичай відбувається у декілька етапів [3]:

1. створення концепції продукту: збір інформації, написання списку задач (product backlog);
2. проєктування архітектури;
3. розробка mock-ups (макетів);
4. написання чи збір контенту *(за вимогою замовника)*;
5. розробка front-end та back-end частин;
6. тестування, огляд та запуск;
7. видання та технічне обслуговування.

На етапі створення концепції продукту необхідно зібрати повну інформацію про застосунок, для цього варто відповісти на наступні питання:

1. Які проблеми має вирішувати застосунок?
2. Що є очікуваним результатом розробки?
3. Яка кількість часу відведена на розробку?
4. На яке фінансування розрахований застосунок?
5. Визначення функціоналу системи.
6. Чи є передбачуваними будь-які обмеження в роботі системи?
7. Які технології будуть використані при розробці?
8. Хто є ЦА застосунку?

Проєктування архітектури застосунку полягає в конструюванні програмного забезпечення, тобто розробку UML-діаграм: варіантів використання, послідовності та кооперації, класів, станів, діяльності,

компонентів, розгортання. Під час розробки використано спеціалізовані продукти, такі як StarUML та Software Ideas Modeler. Створення UML-діаграм допомагає розробнику краще розуміти структуру проєкту.

Після чого варто перейти до розробки макетів застосунку відповідно до поставлених вимог. Даний етап передбачає також створення логотипу, підбір шрифтів та колірної гами застосунку. Під час розробки mock-ups геймифікованого трекеру звичок використано спеціалізований продукт – Balsamiq.

Розробка front-end та back-end частин є наступним етапом реалізації проєкту. Після чого застосунок має бути протестовано та оглянуто на наявність помилок, для уникнення проблем під час роботи ПЗ. Після реалізації застосунок має технічно обслуговуватися під час всього терміну фінансування проєкту.

2.2 Створення Use case

Оскільки Use Case або сценарій користування – це перелік дій, тобто сценарій, відповідно якого користувач буде взаємодіяти із застосунком для виконання будь-якої дії, в залежності від функціоналу додатку. Найчастіше сценарії описують, що система буде виконувати, а не як саме.

Короткий usecase

Користувач має потребу в набутті нової звички, додає її у застосунок з відповідними опціями, крім того користувач потребує допомоги коуча – звертається до нього, отримує відповідні поради та рекомендації. Також користувач має змогу самостійно шукати інформацію у вже підготованих постах та мотиваторах.

Поверхневий usecase

Головний сценарій (успішний):

Користувач має потребу в набутті нової звички, додає її у застосунок з відповідними опціями, крім того користувач потребує допомоги коуча – звертається до нього, отримує відповідні поради та рекомендації. Також

користувач має змогу самостійно шукати інформацію у вже підготованих постах та мотиваторах. Користувач відмічає прогрес набуття звички, виконуючи поставлену ціль (рис. 2.1).

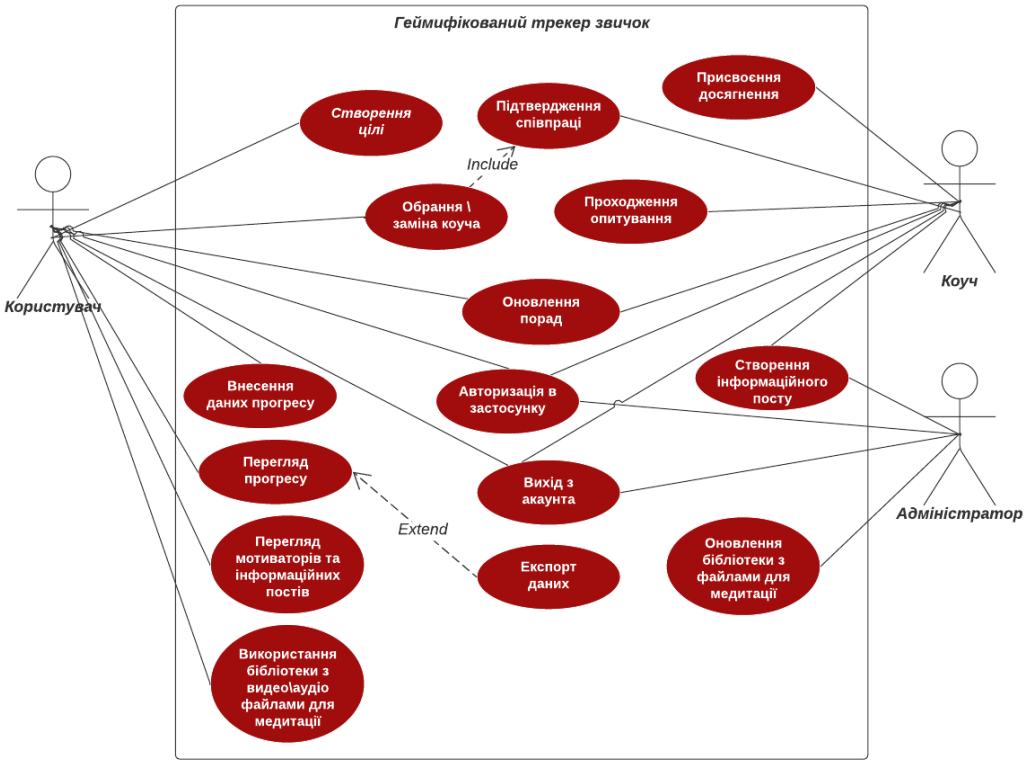


Рисунок 2.1 – Діаграма використання геймифікованого трекеру звичок

Діаграми варіантів використання застосовують для опису функцій високого рівня та області застосування системи (табл. 2.1). Також дані діаграми визначають взаємодії між системою та її акторами, визначають контекст і вимоги або всієї системи, або важливих частин системи. Після розробки короткого та поверхневого сценарія використання, наступним етапом є складання повного usecase, що містить в собі всі деталі використання ПЗ (табл. 2.2).

Таблиця 2.1 – Опис елементів моделі на діаграмах варіантів використання

Елемент	**Опис**
Актори	Актор представляє роль користувача, який взаємодіє з системою, яку ви моделюєте. Користувач може бути користувачем-людиною, організацією, машиною або іншою зовнішньою системою.

Кінець таблиці 2.1

Елемент	Опис
Підсистеми	У моделях UML підсистеми є типом стереотипних компонентів, які представляють незалежні, поведінкові одиниці в системі. Підсистеми використовуються в діаграмах класів, компонентів і варіантів використання для представлення великомасштабних компонентів у системі, яку ви моделюєте.
Зв'язки та відношення	Це зв’язки між елементами моделі. Відношення – тип елемента моделі, який додає семантику до моделі, визначаючи структуру та поведінку між елементами моделі.
Випадки використання	Випадок використання описує функцію, яку виконує система для досягнення мети користувача. Випадок використання повинен давати спостережуваний результат, який є цінним для користувача системи.

Альтернативний сценарій:

1) Поради коуча не допомагають (в певній мірі \ взагалі) користувачу в набутті звички, є необхідність додаткових консультацій.

2) Користувач не справляється у зазначений проміжок часу, необхідність збільшити кількість відведеного часу на звичку.

3) Користувач видаляє звичку, не завершуючи план її набуття.

4) Користувач набуває звички завчасно, через що закриває її виконання раніше відведеного часу.

Таблиця 2.2 – Повний usecase

Primary Actor	Користувач
Scope	Гейміфікований трекер звичок
Level	Мета користувача
Preconditions	Користувач авторизований в застосунку
Stakeholders and interests	1. Адміністратор: Зацікавлений у підтримці застосунку, своєчасного оновлення даних та розміщення\видалення актуальної інформації в застосунку. 2. Коуч: Зацікавлений в допомозі користувачу, розміщення актуальної інформації (пости \ мотиватори) в застосунку.

Продовження таблиці 2.2

Stakeholders and interests	3. Користувач: Зацікавлений в набутті звички(-ок) за найкоротший час, із зручним графіком та підтримці.
Main Success Scenario: 1. Користувач встановлює застосунок з метою набуття нової звички \ позбавлення від старої, реєструється.	
2. Користувач додає необхідні цілі в застосунок. 3. Користувач обирає коуча відповідно із власними вимогами. 4. Коуч підбирає поради та корисну інформацію \ завдання для користувача, щоб подомогти в роботі із поставленими користувачем цілями. 5. Користувач використовує рекомендації коуча. 6. Користувач відмічає прогрес. 7. Користувач має труднощі – звертається до коуча. 8. Коуч допомагає із подоланням труднощів. 9. Адміністратор \ коуч додає потенційно цікаву інформацію для користувача в розділ з мотиваторами та рекомендаціями. 10. Користувач продовжує відмічати прогрес, закінчує процес набуття нової звички або позбавлення від старої непотрібної.	
Result	Користувач набув нову звичку або позбавився від непотрібної.
Extensions:	
1.	1. Користувач не встигає виконати процес набуття нової звички або позбавлення від старої непотрібної : 1.1. Користувач повідомляє про це коучу. 1.2. a. Коуч надає рекомендації щодо збільшення терміну виконання процесу. 1.2.b. Коуч повністю \ частково змінює програму досягнення поставленої цілі. 1.3. Коуч надає поради щодо полегшення набуття\позбавлення звички. 1.4. Користувач дотримується порад та програми. 1.5. Користувач відмічає прогрес. 1.6. Користувач досягає поставленої цілі.
2	2. Програма та поради коуча не допомагають користувачу: 2.1. Користувач вирішує змінити коуча. 2.2. Коуч перестає працювати з користувачем, отримувати доступ до його прогресу в виконанні цілей та інших змін.

Кінець таблиці 2.2

2	2.3. Користувач обирає нового коуча. 2.4. Обраний коуч аналізує поставлені цілі та пропонує користувачу нову програму, інформацію та поради. 2.5.a. Користувачу робота з новим коучем є корисна, користувач із ним продовжує співпрацю. 2.4.b. Користувачу не допомагають поради та програма обраного коуча. Користувач вимушений змінювати коуча.
Special Requirements:	1) Система перекладу тексту на декілька мов (українську, англійську). 2) Адаптивний інтерфейс. 3) Створення умов на робочому місці для продуктивної праці коучів з можливістю швидкого реагування на запит користувача: пристрій (ноутбук. планшет, ПК чи ін.), стабільний доступ до Інтернету.
Frequency of Occurrence	Система може працювати майже безперервно.

2.3 Алгоритм роботи програмного забезпечення

Геймифікований трекер звичок має бути зручним та зрозумілим у використанні, оскільки вебзастосунок розрахований на користувачів різного віку. Користувач повинен мати змогу користуватися ПЗ завдяки смартфону (ОС iOS чи Android) або ПК. Дане програмне забезпечення призначене для автоматизації процесу набуття звички, для чого вебзастосунок містить відповідний функціонал.

Важливо, щоб функціонал залишався актуальним та користувався попитом серед усіх трьох типів користувачів: адміністратор, коуч, користувач. Для відображення алгоритмів чи процесів графічно, зазвичай використовують блок-схеми. В яких позначається кожен етап роботи користувача з системою та варіанти розвитку подій відповідно до обранного користувачем варіанта.

Аналогом блок-схеми можна вважати діаграму діяльності, оскільки вона відображає послідовність дій, яка виконується під час процесу реалізації якогось варіанту використання чи в цілому функціонування системи. Завдяки розробці діаграм діяльності можна продемонструвати процес роботи геймифікованого трекеру звичок, оскільки, навідміну від блок-схем, діаграми діяльності підтримують паралельні процеси. Створено три діаграми діяльності з основними функціями користувача:

1) *Створення нової звички* (рис. 2.2). Діаграма містить такі кроки як: перевірка авторизації на коректність, перевірка наявності звички, чи треба додавати нову звичку, бажання користувача додати коментар.

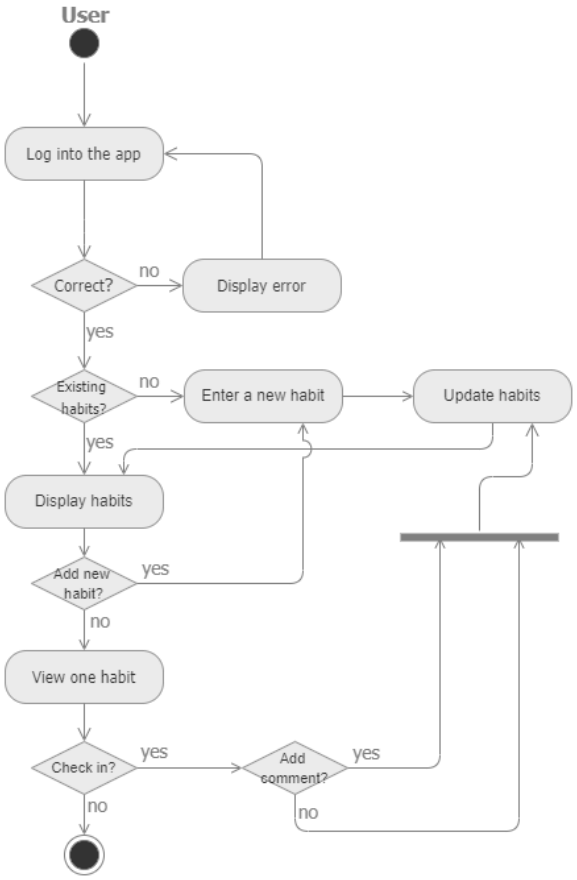


Рисунок 2.2 – Діаграма діяльності створення нової звички

2) *Додавання прогресу* (рис. 2.3). Діаграма містить такі кроки як: перевірка авторизації на коректність, перевірка звички на актуальність (активність), запит чи є ще доповнення по з приводу інших цілей.

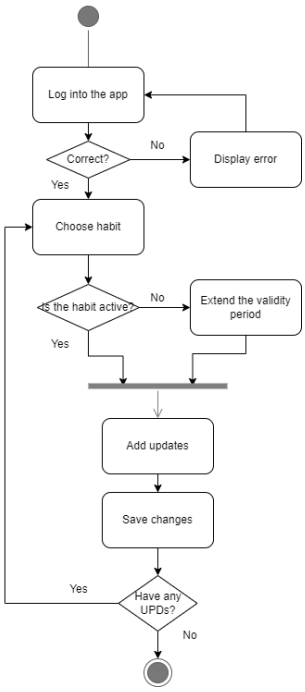


Рисунок 2.3 – Діаграма діяльності додавання прогресу

3) *Перегляду та експорту даних прогресу* (рис. 2.4). Діаграма містить такі кроки як: перевірка авторизації на коректність, необхідність сортування, необхідність експорту даних.

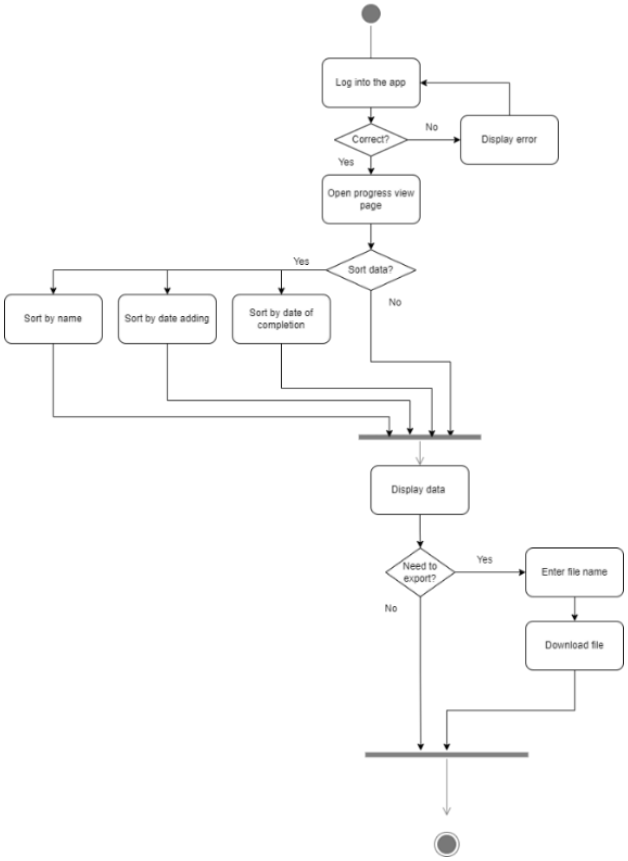


Рисунок 2.4 – Діаграма діяльності перегляду та експорту прогресу

Діаграми діяльності розроблені за допомогою використання спеціалізованого програмного забезпечення Software Ideas Modeler для демонстрації основних процесів роботи користувача з застосунком.

2.4 Розробка діаграми розгортання

Для графічного представлення ПЗ, що розробляється, необхідно створити діаграму розгортання (рис. 2.5), її сенс полягає в тому, щоб відображати обчислювальні вузли, а також компоненти й об’єкти. Даний тип діаграм допомагає змоделювати та продемонструвати фізичне розгортання наявних артефактів, що зосереджені на вузлах.

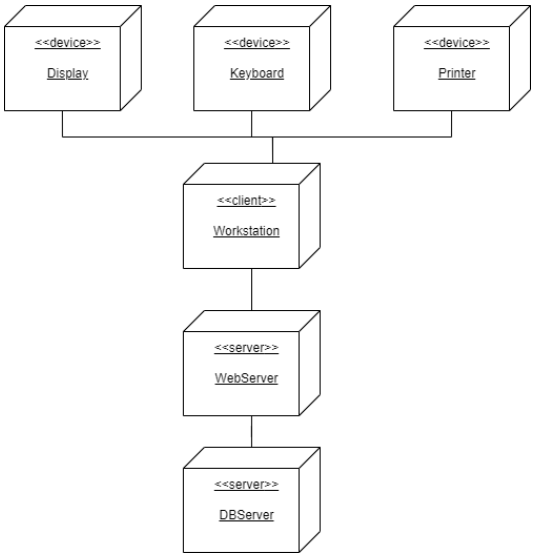


Рисунок 2.5 – Діаграма розгортання геймифікованого трекеру звичок

На діаграмі розортання геймифікованого трекеру звичок вузли представлені у вигляді прямокутних паралелепіпедів, вузли поділяються на два типи: обчислювальні фізичні ресурси (вузол пристрою) та програмні ресурси (вузол середовища виконання). Зазвичай вузол пристрою містить власну пам’ять та забезпечує виконання застосунку за допомогою сервісів, таким чином це можуть бути смартфони, ПК, планшети тощо. Натомість вузол середовища виконання являє собою сервіс, який має змогу використовувати інші програмні елементи, а також працює саме у зовнішньому вузлі. Діаграма даного типу розроблена задля раціональної організації компонентів, оскільки це є важливою

частиною розробки ПЗ, адже саме це впливає на продуктивність та безпеку системи, що розроблюється. Діаграма розгортання геймифікованого трекеру звичок містить у собі комплекс з тришаровою архітектурою (табл 2.3).

Таблиця 2.3 – Опис архітектури програмного комплексу

№ шару	Компоненти	Опис
1	Client – workstation. Devices: – display; – keyboard; – printer.	*Робоча станція* (workstation) з пристроями введення та виведення (device).
2	Server – webserver.	*Вебсервер*, який виконує основну бізнес-логіку застосунку.
3	Server – DBServer.	З вебсервера проходить доступ до *сервера баз даних*.

2.5 Побудова та використання діаграм взаємодії

Під час аналізу та моделювання системи важливо розуміти як користувач будь-якого рівня буде взаємодіяти із застосунком. Діаграма №1 (рис. 2.6) відповідає за демонстрацію взаємодії користувача із засосунком при додаванні нової звички, демонструє інтерфейси, параметри та методи для виклику в результаті взаємодії користувача із застосунком з ціллю додавання нової звички.

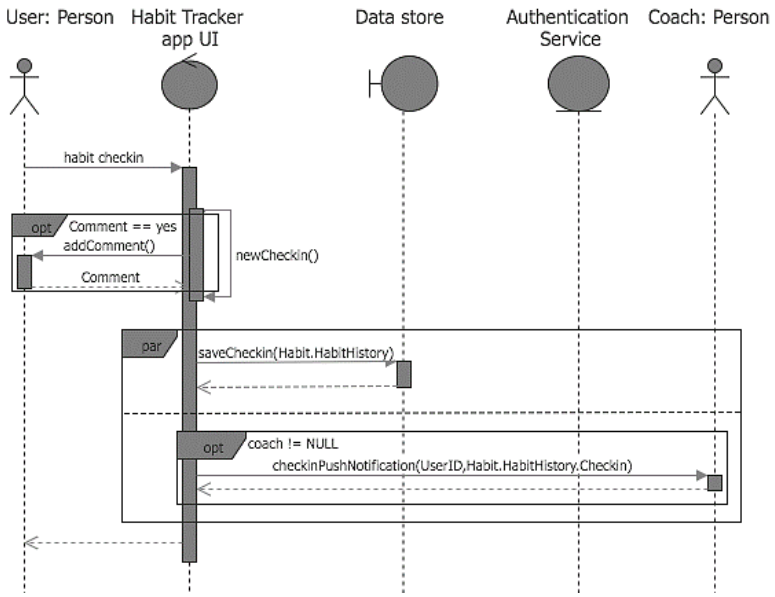


Рисунок 2.6 – Діаграма взаємодії додаванні нової звички

Діаграма №2 (рис. 2.7) відповідає за демонстрацію взаємодії користувача із засосунком для показу історії звичок. На ній зображено які саме інтерфейси (Habit Tracker app UI, Data Store), параметри (opt) та методи (par) буде викликано в результаті взаємодії користувача із застосунком з ціллю перегляду історії звичок.

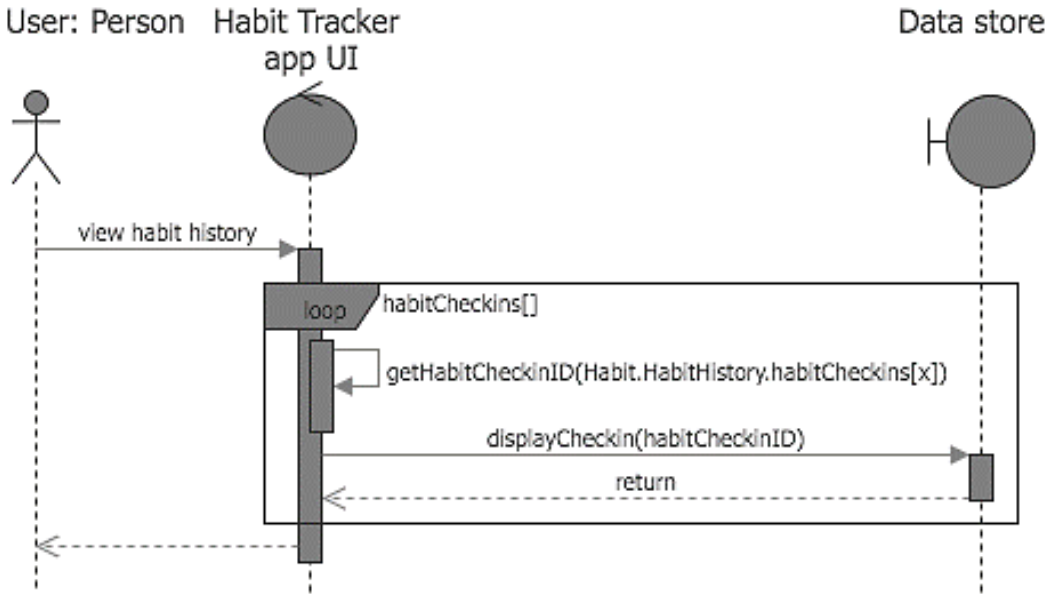


Рисунок 2.7 – Діаграма взаємодії показу історії звичок

Остання, тобто діаграма №3 (рис. 2.8) відповідає за демонстрацію одразу декількох варіантів взаємодії користувача із засосунком: авторизація, реєстрація користувачів, додавання нової звички, відображення та імпортування історії звичок. На ній зображено які саме інтерфейси (Habit Tracker app UI, Data Store, Autentification Service), параметри (opt) та методи (par) буде викликано в результаті взаємодії користувача із застосунком з даними цілями.

Крім того діаграма містить скорочені версії попередніх діграм (рис. 2.6-7), для демонстрації роботи застосунку при використанні різних можливостей функціоналу гейміфікованого трекеру звичок «Habitracker», так можна розглянути повну роботу трекеру з початку його запуску (start app) до повернення застосунку в початковий стан.

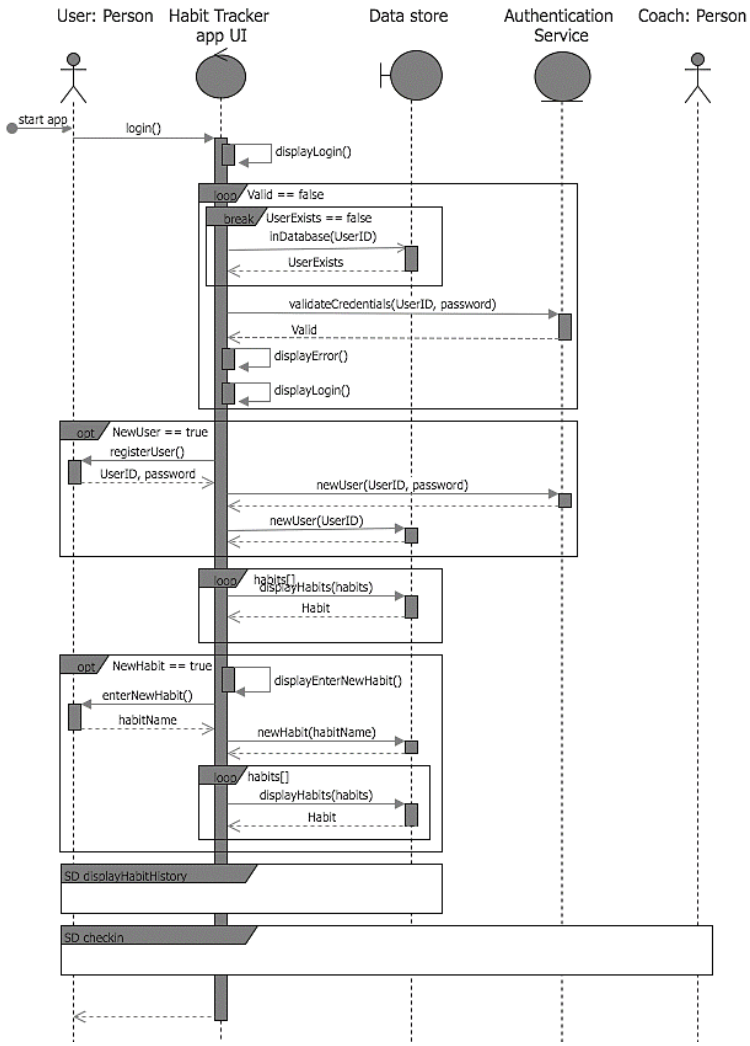


Рисунок 2.8 – Діаграми взаємодії реєстрації, показу історії звичок, додавання нової звички, додавання користувача, авторизації

2.6 Розробка шаблону MVC для керованого інтерфейсу

Останнім етапом моделювання геймифікованого трекеру звичок є створення архітектурного патерну MVC [4]. Загальний вигляд шаблону (рис. 2.6) містить в собі три компоненти (табл. 2.4), які взаємодіють між собою.

Таблиця 2.4 – Опис компонентів шаблону MVC

Назва	**Роль**
Модель даних (model)	Демонстрація поведінки ПЗ.
Інтерфейс користувача (view)	Демонстрація інформації (наприклад, гістограма, графік тощо), яку буде отримано на виході.

Кінець таблиці 2.4

Назва	Роль
Модуль керування (controller)	Отримання вхідних данних, а також їх перетворення на команди як для моделі даних, так і для інтерфейсу користувача.

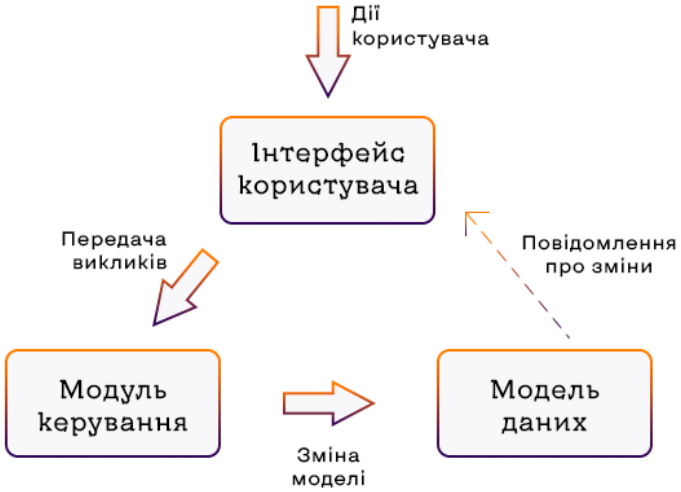


Рисунок 2.6 – Загальний вигляд патерну MVC

Робота MVC патерну геймифікованого трекеру звичок проходить у декілька етапів (табл. 2.5). Етапи виконання патерну графічно відображені на рис. 2.7, куди додано певного користувача, який виконує дії.

Рисунок 2.7 – MVC-патерн геймифікованого трекеру звичок

Таблиця 2.5 – Опис етапів процесу роботи патерну MVC

№ етапу	Опис
1	Користувач відвідує (заходить) на вебресурс, після чого спрацьовує відповідний скрипт ініціалізацій, який зі свого боку має створити екземпляр застосунку, після чого запустити та виконати його. В цей час для користувача завантажується інтерфейс головної сторінки вебзастосунку з усіма відповідними атрибутами [5].
2	При отриманні запиту від користувача, застосунок має визначити контролер, до якого надійшов запит, а також саму дію. За умови завантаження головної сторінки, буде виконана дія, що встановлена за замовчуванням, тобто index.
3	Застосунок має створити певний екземпляр визначеного контролера, після чого запустити метод дії. Наприклад, звернення користувачем до моделі, яка виконує CRUD-операції стосовно данних в БД.
4	Останній етап – отримання даних з моделі та виведення результатів користувачу.

Висновки до розділу 2

У другому розділі з моделювання геймифікованого трекеру звичок, описано етапи реалізації вебзастосунку, а також алгоритми роботи ПЗ та вирішення поставленої задачі. Завдяки чому закріплені навички у побудові та використанні діаграм взаємодії, діяльності та розгортання, у розробці та застосуванні шаблонів MVC. На основі діаграми використання складено сценарії використання, після розробки короткого та поверхневого сценарія використання, складено повного usecase, що містить в собі всі деталі використання ПЗ.

Для моделювання інтерфейсів створено архітектурний патерн MVC, який графічно відображає процес взаємодії користувача із геймифікованим трекером звичок, на основі загального вигляду патерну. Використані UML-діаграми в даному розділі чітко та виразно демонструють структуру та вимоги до застосунку, а також описують архітектуру програмного комплексу.

3 ПРОЄКТУВАННЯ ВЕБЗАСТОСУНКУ ТА ОГЛЯД СТЕКУ ЗАДІЯНИХ ТЕХНОЛОГІЙ

3.1 Розробка UML-діаграм

Зазвичай для моделювання та проєктування ПЗ використовують діаграми, найпоширенішими та досить зручними є UML-діаграми, оскільки завдяки ним можна описати бізнес-процеси, системне проєктування, а також відобразити різні організаційні структури. UML налічує 12 видів діаграм (табл. 3.1).

Таблиця 3.1 – Різновид та опис UML-діаграм

Тип діаграм	Назва	Використання
Структурні	Діаграма класів	При моделюванні структури застосунків об’єктно-орієнтованих, включаючи їх класи, атрибути, назви методів, наслідування та зв’язків між ними [6].
Структурні	Діаграма компонентів	При моделюванні структури компонентів застосунку.
Структурні	Діаграма об’єктів	При моделюванні окремої частини працюючого застосунку, для відображення реально існуючих в поточний час екземплярів класів, включаючи значення їх атрибутів [7].
Структурні	Діаграма композитних структур	При моделюванні складових структурних елементів моделей.
Структурні	Діаграма розгортання	При моделюванні апаратної частини системи, з якою взаємодіє ПЗ або розташоване.
Структурні	Діаграма пакетів	Моделювання частин об’ємних моделей.
Поведінкові	Діаграма активностей	Складання специфікації бізнес-процесів та алгоритмів, що має автоматизувати ПЗ.
Поведінкові	Діаграма використання	Формування вимог, відповідно до бажань користувача, замовника тощо.
Поведінкові	Діаграма кінцевих автоматів	Формування поведінки реактивних систем.
Поведінкові (взаємодії)	Діаграма послідовностей (включаючи тимчасові діаграми)	Моделювання тимчасових поглядів стосовно зовнішніх та внутрішніх протоколів програмного забезпечення.
Поведінкові (взаємодії)	Діаграма схем взаємодії	Формування ієрархії діаграм послідовностей
Поведінкові (взаємодії)	Діаграма комунікацій	Моделювання тимчасових поглядів стосовно зовнішніх та внутрішніх протоколів програмного забезпечення у вигляді графів. Аналог діаграми послідовностей.

Під час етапу розробки – моделювання геймифікованого трекеру звичок, у другому розділі, використано діаграми взаємодії, діяльності та розгортання оскільки при моделюванні системи саме ці діаграми найбільш чітко та виразно відображається структура та вимоги до застосунку. Після чого, на етапі проєктування використано діаграми класів, станів та переходів, компонентів та пакетів, оскільки дані діаграми відображають поведінку, архітектуру ПЗ та його апаратної частини. Для створення діаграм використано застосунок StarUML.

3.1.1 Діаграма класів

Діаграма класів (рис. 3.1) є основою для об’єктно-орієнтованого моделювання, а саме вона використовується зазвичай для загального і детального проєктування структури програми, також для переведення моделей безпосередньо у програмний код. Програма має необхідний базовий функціонал (компоненти) для роботи з діаграмами класів. Кожен з класів (табл. 3.2) має власну назву, яка має пояснювати, за що відповідає клас, а також методи та параметри, що входять до цього класу.

Таблиця 3.2 – Класи та їх характеристика

Клас	**Компоненти**	**Методи**	**Призначення класу**
Person	ID	newPerson()	Додавання нового користувача
	Name		
User	Habits	NewUser()	Клас користувача
	HabitNum		
	Coach	DisplayHabits()	
	Users		
Coach	Users	NewCoach()	Клас коуча
		DisplayAllUsers()	
		AddUser()	
		DeleteUser()	
		ViewUserHabbits()	
		CheckinPushNotification()	

Кінець таблиці 3.2

Клас	Компоненти	Методи	Призначення класу
Habit	HabitId	NewHabit()	Клас звички(-ок)
	HabitName	DisplayHabit()	
	HabitHistory	DeleteHabit()	
		CheckIn()	
HabitHistory	HabitId	NewCheckins()	Клас історій звичок
	HabitCheckins	DisplayHistory()	
CheckIn	HabitCheckinID	AddComment()	Клас створення звичок та коментарів
	CheckinDate	DisplayCheckin()	
	Comments	AddComment()	
Comment	CommentID	DisplayComment()	Клас коментарів
	CommentText		
	CommentDate		
	Commenter		

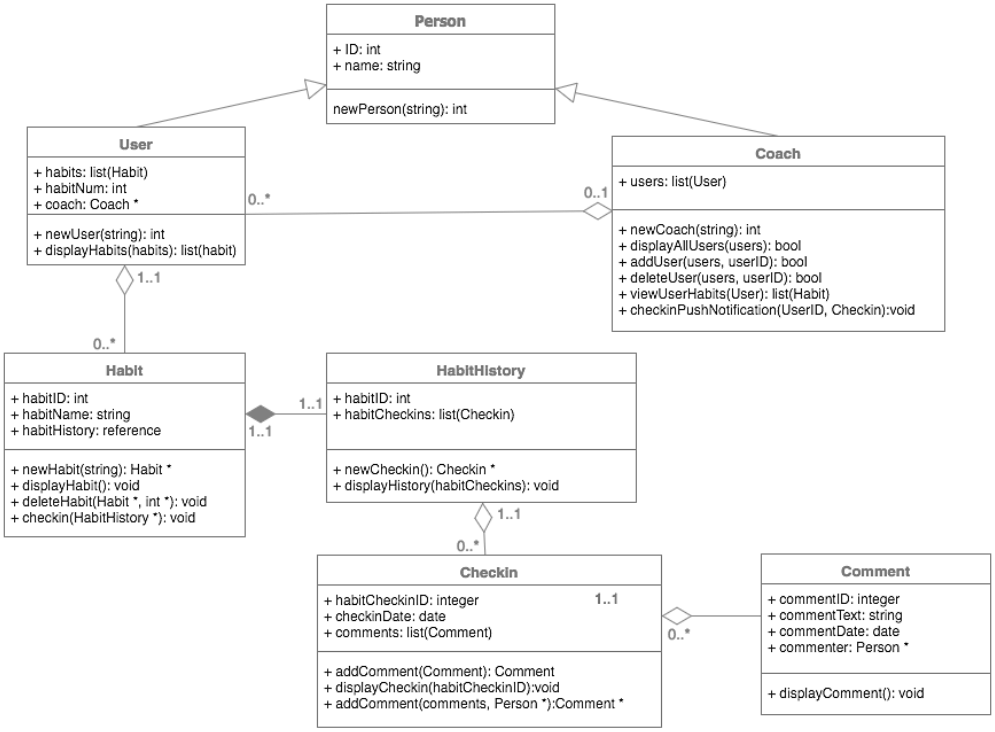


Рисунок 3.1 – Діаграма класів

Створена діаграма має наступні особливості:

1. Діаграма класів відображає лише взаємодію між користувацькими класами, що містять логіку застосунку. Стандартні класи фреймворку не було додано на діаграму.

2. Діаграма єдина для всіх класів, тобто вони всі пов’язані між собою, тому не варто розривати зв’язки між ними.

3. Певні типи зв’язків не представлені на діаграмі, оскільки їм важко знайти логічне застосування в межах цього застосунка.

3.1.2 Діаграми станів та переходів

Statechart diagrams або діаграми станів зазвичай демонструють певні сценарії, які будуть виконуватися під час функціонування системи [8]. Так для вебзастосунку геймифікованого трекеру звичок розроблено 3 діаграми станів та переходів:

1) для всього застосунку (рис. 3.2), яка демонструє можливі дії користувача та реакцію застосунку на них;

2) для здійснення CRUD-операцій над цілями (рис. 3.3);

3) для здійснення додавання цілі та обрання коуча (3.4).

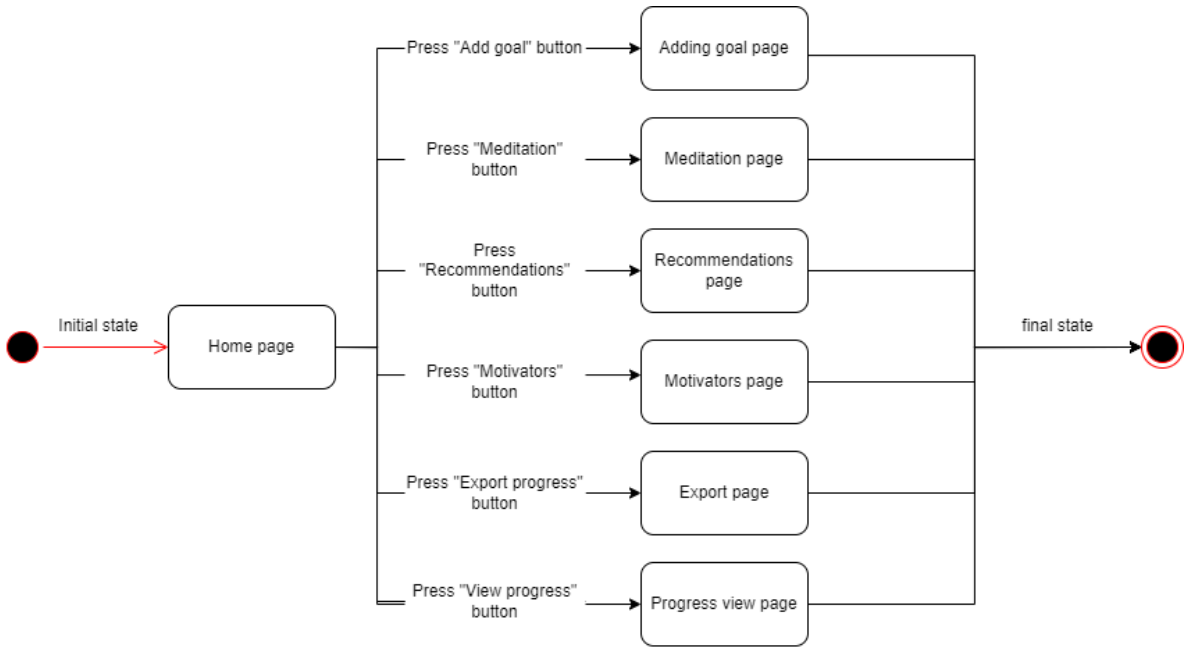


Рисунок 3.2 – Діаграма станів та переходів з головної сторінки до інших з меню

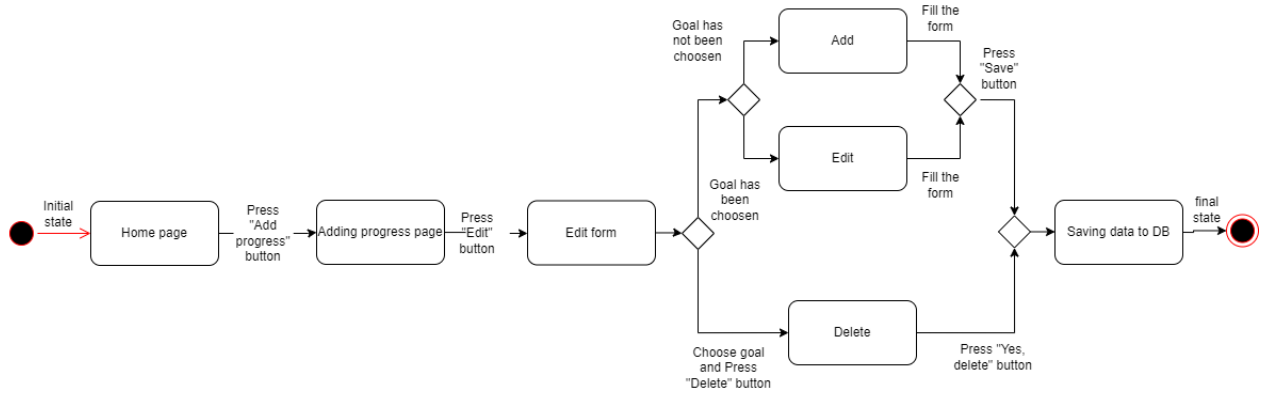


Рисунок 3.3 – Діаграма станів при зміні чи видаленні звички

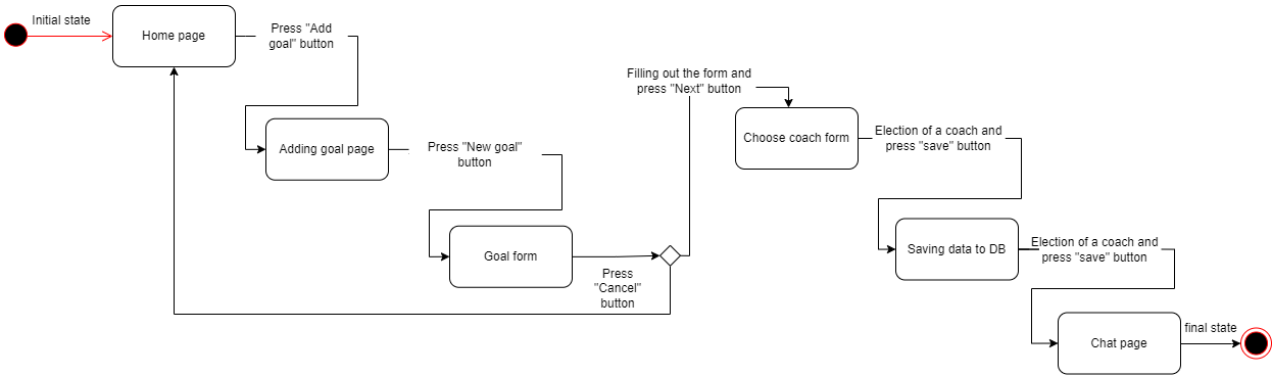


Рисунок 3.4 – Діаграма станів при додаванні цілі та обиранні коуча

Таким чином, головним призначенням даних діаграм є опис основних можливих послідовностей станів та переходів, що в сполучені протягом всього життєвого циклу модельованої системи мають характеризувати її поведінку. Крім того, важливо зазначити, що при перебуванні об’єкта чи модельованої системи в початковому стані, її поведінка може супроводжуватися реалізацією певних внутрішніх дій [9]. Після чого зміна стану об’єкта з поточного буде можливою лише у 2-х випадках:

1. після завершення дій, що виконуються;
2. при виникненні певних зовнішній подій.

3.1.3 Діаграми компонентів

Component diagram знаходять застосування в моделюванні фізичних аспектів різноманітних об’єктно-орієнтованих систем, що зі свого боку можуть використовуватися як для визначення, візуалізації та документування будь-яких

компонентних систем, крім того їх можна використовувати для побудови систем, що виконуються, за допомогою як прямого, так і зворотного проєктування [10]. Проте, щоб створити діаграми компонентів та розгортання необхідно розділити діаграму класів, на 3 шари (рис. 3.5).

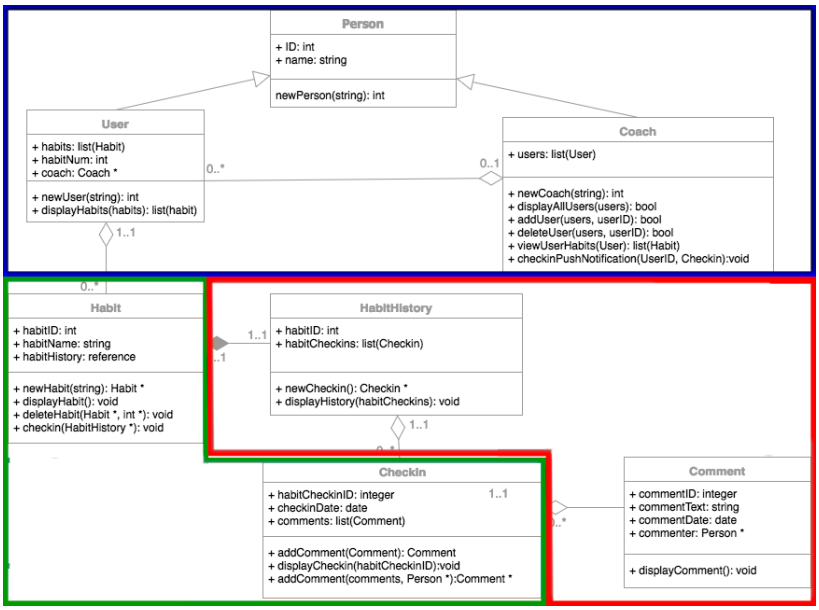


Рисунок 3.5 – Діаграма класів з тришаровою архітектурою

Перший шар (позначений синім кольором в лівій частині діаграми) складається з сукупності класів, які формують інтерфейс користувача. До нього належить ієрархія діалогів системи та клас прав користувача, що використовується в діалозі авторизації. Component

Другий шар (обмежений лінією зеленого кольору) відповідає за бізнес-логіку (перелік алгоритмів, що оперують даними для виконання функцій системи). Він сформований сукупністю класів алгоритмів та класом прав користувача на виконання тих чи інших операцій.

Третій шар (обмежений лінією червоного кольору) забезпечує обмін даними з базою даних. Він складається зі специфічних структур даних, кожна з яких відповідає рядку відповідної таблиці бази даних, контейнерів структур даних, класу інтерфейсу бази даних та прав доступу до таблиць БД.

Відповідно до діаграми класів з тришаровою архітектурою складено діаграму компонентів для гейміфікованого трекеру звичок (рис. 3.6).

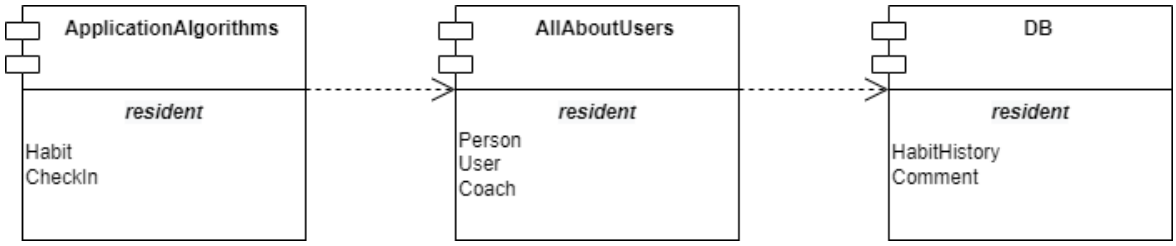


Рисунок 3.6 – Діаграма компонентів

На діаграмі зображено 3 компонента (зліва – направо):

Перший компонент – «ApplicationAlgorithms» має 2 класи: Habit та CheckIn. Клас пов’язаний відношенням направленої асоціації з компонентом «AllAboutUser» (тобто залежить від нього), тому використано відношення залежності – штриховану стрілку.

Другий компонент – «AllAboutUser» містить класи Team, WorkerList, List, Worker, які приймають данні про працівників, їх інформацію, які існують команди на підприємстві, їх учасники. Дані беруться с компоненту «DB».

Третій компонент – «DB» має стереотип database, та класи HabitHistory, Comment, які містять всі необхідні дані.

3.1.4 Діаграма пакетів

Package diagram (рис. 3.7) необхідна для елементів у групі за будь-яким визнанням з метою спрощення структури та організації роботи з модельною організацією системи. Пакет – це інструмент групування, що дозволяє взяти конструкцію UML і об’єднати її елементи в одиницях високого рівня [11].

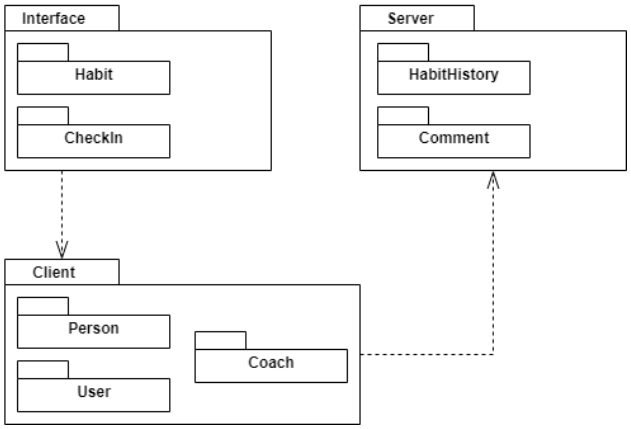


Рисунок 3.7 – Діаграма пакетів

– Пакет «Interface» має 2 класи: Habit та CheckIn. Клас пов’язаний відношенням направленої асоціації з компонентом «Client» (тобто залежить від нього), тому використано відношення залежності – штриховану стрілку.

– Пакет «Client» містить класи Team, WorkerList, List, Worker, які приймають данні про працівників, їх інформацію, які існують команди на підприємстві, їх учасники. Дані беруться с пакету «Server».

– Пакет «Server» має стереотип database, та класи HabitHistory, Comment, які містять всі необхідні дані.

3.2 Оглядання технологій

При розробці вебзастосунку «Геймифікований трекер звичок» використано відповідний стек технологій (рис. 3.8):

– мови програмування: PHP та JavaScript;

– технології front-end розробки: Bootstrap та Vue.js;

– технології back-end розробки: Laravel та MySQL.

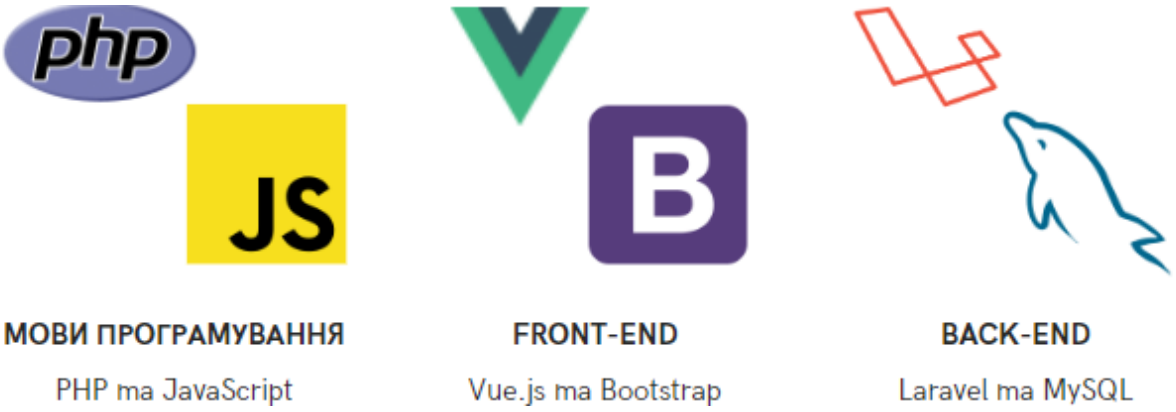


Рисунок 3.8 – Стек технологій з логотипами відповідно

Вибір стеку технологій досить важливий, оскільки від нього залежить продуктивність роботи застосунку, а також формування вимог до апаратних частин пристрою та надійність роботи ПЗ. Для кожної задачі необхідно підібрати відповідний інструмент її реалізації. Відовідно до бізнес-процесів геймифікованого трекеру звичок, обрано необхідний інструментарій, що досить часто використовується при розробці вебзастосунків.

3.2.1 Мови програмування

Однією з основних та досить популярних мов програмування є *PHP* або Hypertext Preprocessor. Оскільки вебсайти, що написані цією мовою, вважаються більш гнучкими. Основними перевагами мови є безпека, швидкість обробки сценаріїв та актуальність, що змушує її оновлюватися та вдосконалюватися [12]. Крім того, PHP має багато безкоштовних фреймворків: Laravel, Yii, Symfony тощо. Найпоширенішим для веброзробки є фреймворк *Laravel* [13]. Також PHP підтримує досить велику кількість БД, включаючи вільну систему для керування реляційними БД – *MySQL*, яка спроможна швидко оброблювати запити у великих БД. Тобто дана мова програмування у зв’язку з БД та використанням фреймворку Laravel має утворювати структуру back-end розробки [14]. Важливим фактором є те, що PHP має змогу поєднуватися та працювати з іншими мовами програмування, такими як, в даному випадку, *JavaScript*.

Зі свого боку JS є динамічною скриптовою мовою програмування високого рівня, також вважається основною мовою веброзробки, але більше підхожа саме для частини front-end розробки. Особливістю мови є унікальна підтримка усіма ОС, стабільна робота у всіх браузерах та на мобільних платформах. JS працює у зв'язці з HTML та CSS, а також може працювати без підключення до мережі Інтернет. Як і PHP, має багато безкоштовних фреймворків для використання, серед яких *Vue.is* та *Bootstrap*, які значно спрощують процес реалізації розробки front-end частини.

Для розробки геймифікованог трекеру звичок використано безкоштовний текстовий редактор Atom з відкритим для всіх вихідним кодом. Перевагу віддано даному текстовому редактору, оскільки його головною перевагою є наявність потужного інтрументарію, який повністю відповідає потребам розробника, а також можливості змінити і налаштувати під себе все, що завгодно. Наприклад, можна змінити навіть зовнішній вигляд редактора, лише за умови знання JavaScript.

3.2.2 Технології front-end розробки

Vue.is та Bootstrap, як зазначено в попередньому підрозділі, є фреймфорками JavaScript. Vue.js використовується для реалізації користувацького інтерфейсу у вебзастосунках та вебсайтах. Відповідно до опитувань компанії Monterail станом на 2021 рік [15], зроблено висновки, що Vue.js користується попитом та, за прогнозами опитаних компаній та розробників, буде успішно використовуватися у майбутньому. Оскільки фреймворк має унікальні переваги та з кожним роком, відповідно до порівняльної статистики з минулими роками, набуває все більше користувачів. У опитуванні компанії Monterail з грудня 2020 року до січня 2021 року (табл. 3.3), кількість опитуваних склала 1635 осіб.

Таблиця 3.3 – Ключова інформація згідно опитування [15]

%	Теза
90	Висока ймовірність використання Vue.js для свого наступного проєкту.
93	Використовували офіційну документацію, для отримання інформації стосовно роботи з Vue.js.
76	Відмітити чіткосформовану та доступну документацію як на найбільшу перевагу фреймворка.
60	Почали використовувати під час розробки, оскільки фреймворк є досить простим для початківців.
56	Згодні з прогнозом, що Vue.js стане ще більш популярним у їхніх організаціях протягом наступних 12 місяців.

Зі свого боку фреймворк Bootstrap допомагає у верстанні вебсайтів більш зручно та швидко, ніж при використанні «чистих» CSS та JS. Основною перевагою є доступність, оскільки навіть початківцю у веброзробці буде досить комфортно та зрозуміло створювати якісні макети. На рівні структури Bootstrap являє собою набор файлів CSS та JS, в яких є основні інструменти даного

фреймворка: сітка Bootstrap, відповідні класи й компоненти [16]. Також суттєвими перевагами є:

– наявність великої кількості навчальних матеріалів та добре структурована документація;

– наявність ліцензії MIT, яка забезпечує безкоштовне використання фреймворку як і у відкритих, так і в комерційних проєктах;

– однорідність дизайну та його узгодженість між компонентами;

– кросбраузерність та кросплатформність;

– можливість налаштування змінних за бажанням та вимогами проєкту.

3.2.3 Технології back-end розробки

Для розробки back-end частини використано повністю безкоштовний PHP-фреймворк Laravel з відкритим кодом, який допоможе з визначенням структури застосунку та буде слугувати її каркасом. Основне призначення даного фреймворку – розробка на основі Model-View-Controller. Також він дозволяє спростити рішення таких задач: авторизація, маршрутизація, сесії, кешування тощо. Згідно загальносвітової статистики від Google Trends (рис. 3.9), фреймворк Laravel займає впевнене перше місце і постійно набирає популярність [17]. Станом на 11 червня 2022 року, як в статистиці світовій, так і в Україні (рис. 3.10), даний фреймворк є безперечним лідером в порівнянні з іншими PHP-фреймворками: CodeIgniter, Symfony, Yii, CakePHP.

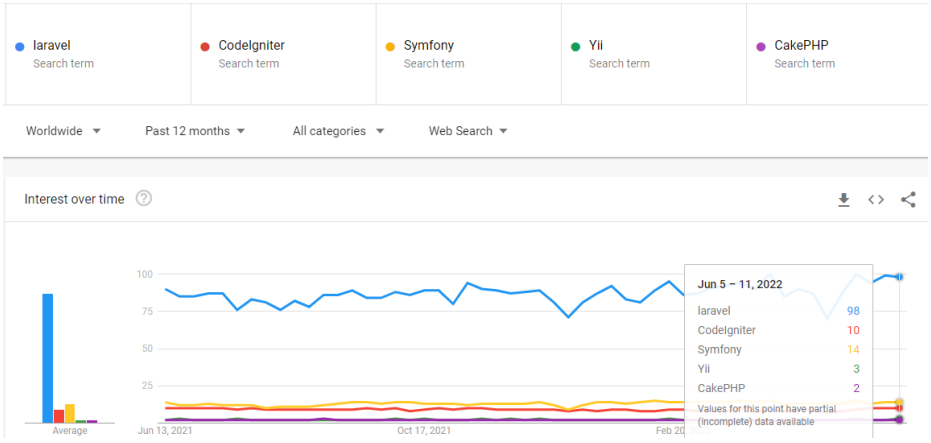


Рисунок 3.9 – Статистика використання фреймворків в світі від Google Trends

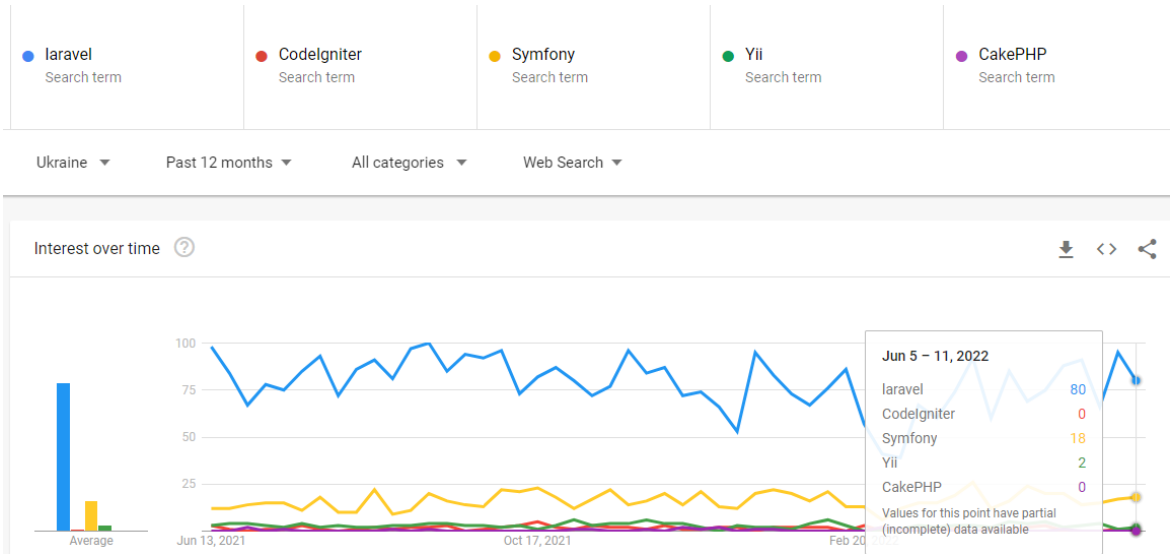


Рисунок 3.10 – Статистика використання фреймворків в Україні від Google Trends

Вагомі переваги Laravel:

– велика кількість мануалів (інструкцій) та інформації для користувача від офіційних розробників;

– логічно структурована та якісно оформлена документація;

– наявність ліцензії MIT, яка забезпечує безкоштовне використання фреймворку як і у відкритих, так і в комерційних проєктах;

– наявність шаблонізатору blade.

Основною причиною вибору даного фреймворку є його універсальність, його можна застосовувати як і для нестандартних проєктів, так і для тих, що будуть розширятися. Крім того, фреймворк досить потужний і гнучкий з обширними можливостями, оскільки містить в собі велику кількість попередньо встановлених бібліотек, які є простими та доступними для розробника. Також завдяки об’єктно орієнтованості Laravel можна використовувати код повторно, запобігаючи його дублювання та, тим самим, перенавантаження коду [18].

Для зберігання та обробки даних необхідно використовувати БД. Бази даних існують 2-х типів: реляційні та нереляційні [19]. Хоча, на перший погляд, нереляційні БД можуть здаватися більш зручними через високу швидкість обробки даних, дані БД мають суттєві недоліки у вигляді прив’язки до

конкретної СКБД та обмеження ємності вбудованої мови запитів.

Реляційні БД, зі свого боку, не мають цих проблем, оскільки мова SQL є досить універсальним, що дає можливість змінювати СКБД без переписування всього коду, оскільки в реляційних БД відсутня жорстка прив'язка до СКБД. Обмеження ємності вбудованої мови запитів в реляційних БД також відсутнє, через велику кількість стандартів SQL з обширним інструментарієм для операцій з даними. Крім того, через те, що більша кількість мов запитів та методів API нереляційних БД ґрунтується на функціях SQL – вони схожі, але NoSQL (нереляційна БД) має меншу функціональність [20]. Через що вирішено використовувати саме реляційний тип баз даних, серед безкоштовних основними є MySQL та PostgreSQL, їх різниця в тому, що основний напрям MySQL – висока ефективніть та продуктивність, а PostgreSQL – велика кількість налаштувань і стандартів. Для розробки геймифікованого трекеру звичок відсутня необхідність суттєвих налаштуваннях БД, краще зробити акцент саме на підвищенні ефективної роботи з БД.

Висновки до розділу 3

Основою третього розділу є проєктування геймифікованого трекеру звичок, а також огляд стеку використаних технологій: мов програмування (PHP, JS), технології front-end (Bootstrap, Vue.js) й back-end (Laravel, MySQL) розробки. Під час проєктування вебзастосунку закріплено навички із розробки UML-діаграм: класів, станів та переходів, компонентів, пакетів. Кожна із створених діаграм детально, водночас лаконічно описана, що спрощує подальшу роботу із ними.

При огляді стеку технологій, проведено аналіз та порівняння актуальних мов програмування вебзастосунків, їх фреймворків, а також видів БД. Завдяки результатам проведеного аналізу визначено технології для розробки front-end та back-end частин. Обґрунтовано вибір використання програмних засобів для реалізації геймифікованого трекеру звичок.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ ТА ПРОВЕДЕННЯ ТЕСТУВАННЯ

Вебзастосунок геймифікований трекер звичок «HabiTracker» містить в собі 7 сторінок (головна, коуч, досягнення тощо) для авторизованого користувача. Систему, після етапу реалізації, можна розмістити для роботи в локальній мережі, а також в мережі Інтернет, після чого з’явиться можливість надавання доступу користувачам. Вебзастосунок створено для 3-х типів користувачів: клієнт (основний користувач), коуч, адміністратор. Ролі клієнта та адміністратора досить схожі, відповідно створених use cases, а основним користувачем системи є клієнт. Відповідно сформованих вимог та функціоналу системи, користувач має можливість виконувати CRUD-операції стосовно власних звичок, оновлення їх прогрес-бару, переглядати отримані від коуча поради, досягнення та методичні вказівки, а також використовувати інтегровану в геймифікований трекер звичок бібліотеку матеріалів для медитації.

4.1 Огляд дизайну вебзастосунку

Для створення mock-ups використано спеціалізований застосунок «Balsamiq Wireframes». Для не авторизованого в системі користувача доступна головна сторінка (рис. 4.1) з базовою інформації стосовно геймифікованого трекеру звичок, а також кнопки для реєстрації (рис. 4.2) або входу в систему.

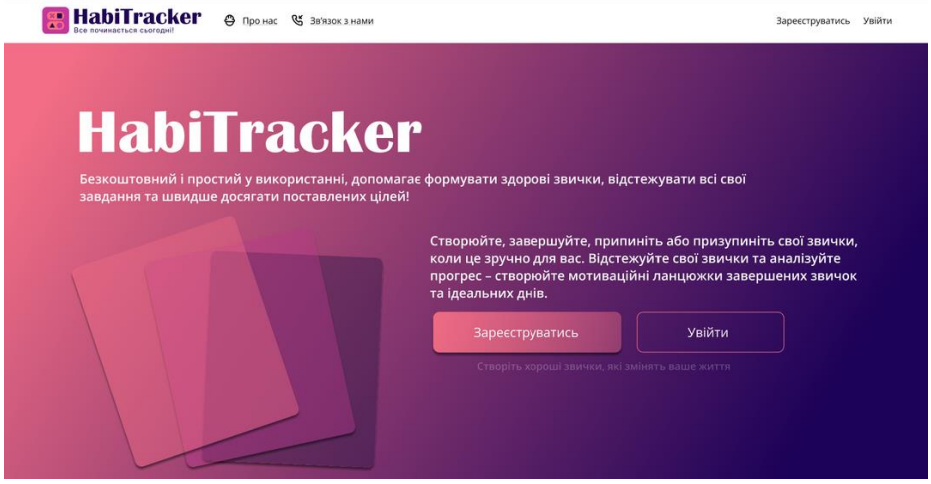


Рисунок 4.1 – Головна сторінка вебзастосунку

Окрім основних функцій користувачу доступні кнопки для відображення інформації про застосунок, а також контакти для звя’зку з розробниками чи компанією, у випадку виникнення проблемних стуацій, запитань чи пропозицій.

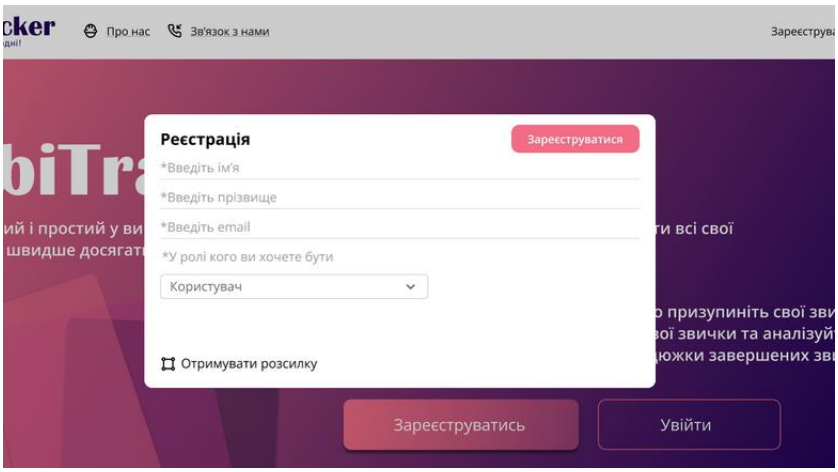


Рисунок 4.2 – Вікно реєстрації користувача

На сторінці створення облікового запису (реєстрації) користувачу необхідно ввести власні дані, такі як: ім'я, прізвище, електрону пошту, пароль. Крім того важливо можна підписатися на розсилку новин застосунку та нагадувань стосовно звичок, для цього треба поставити галочку у відповідному місці. Також розміщено логотип застосунку, який водночас є кнопкою з посиланням на головну сторінку.

Для авторизованого користувача доступно 6 сторінок в меню (рис. 4.3):

– цілі – додавання (рис. 4.4), редагування та видалення цілей, відмічання прогресу;

– коуч – перегляд наданих порад;

– досягнення – перегляд отриманих досягнень;

– медитація – використання бібліотеки матеріалів для медитації (рис. 4.6) з відповідним фільтрацією та сортуванням;

– методичні вказівки – перегляд базових методичних вказівок;

– прогрес – перегляд прогресу (рис. 4.5) стосовно виконання поставлених цілей.

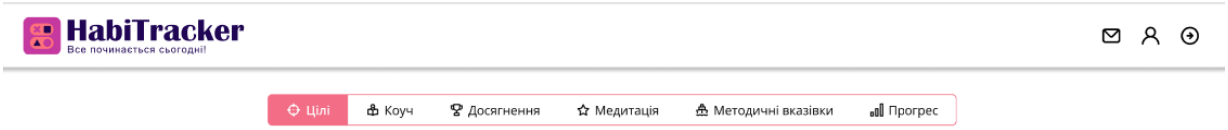


Рисунок 4.3 – Меню вебзастосунку

На головній сторінці для авторизованого користувача розміщено меню з основними розділами: цілі, коуч, досягнення, медитація, рекомендації та прогрес. Також з цієї сторінки перейти до налаштувань, підтримки, вийти з облікового запису чи переглянути його.

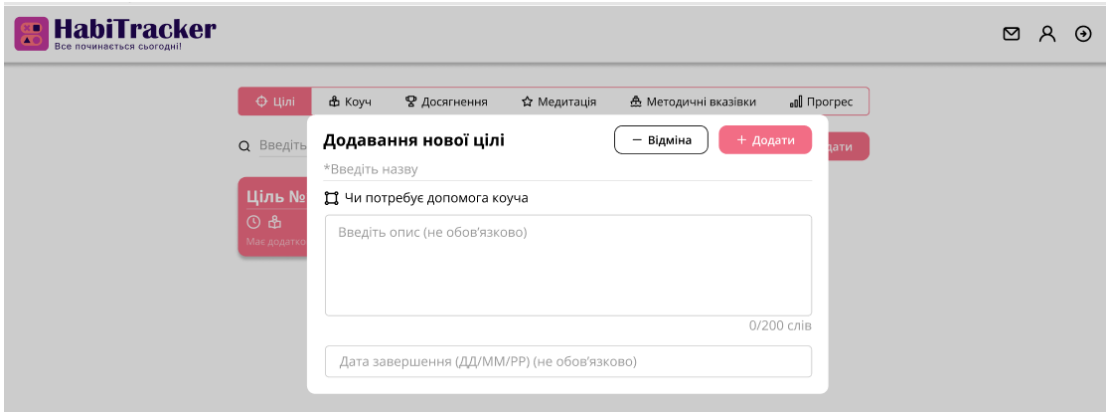


Рисунок 4.4 – Mock-up додавання звички

Сторінка додавання звичок має відповідний функціонал для створення нової звички. Окрім логотипу та базового верхнього меню, на сторінці розміщені поля для додавання назви та детального опису звички, цифровий степпер для вказування кількості днів, світч з вімкненням або вимикненням нагадування та перемикач для вказання необхідності допомоги коуча.

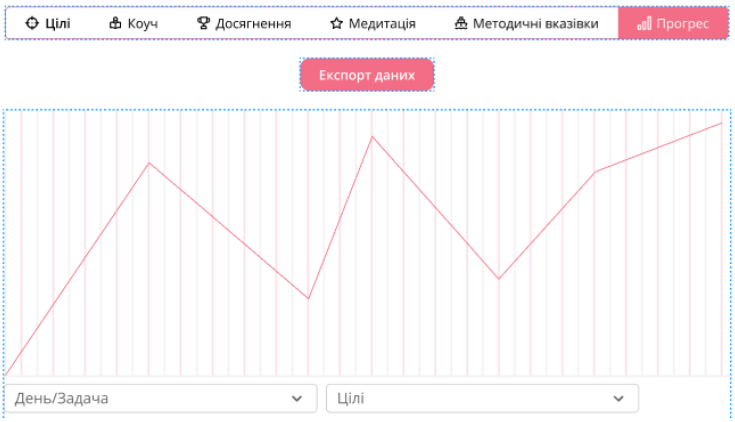


Рисунок 4.5– Mock-up відображення прогресу

Перегляд прогресу здійснюється на відповідній сторінці, на якій розміщено основні відомості про звичку: назва, детальний опис, загальна кількість днів та потреба в допомозі коуча. Крім цього застосунок показує кількість днів, що залишилися та загальний прогрес на прогрес барі. Також можна на кнопку «Експорт» можна завантажити звіт в який буде включено також досягнення (отримані), що пов’язані з даною звичкою.

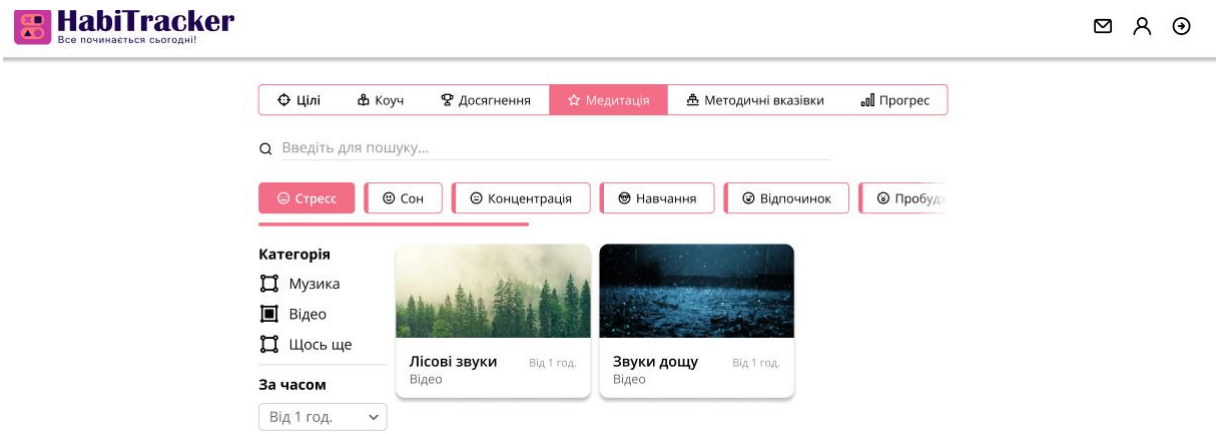


Рисунок 4.6 – Mock-up сторінки медитацій

Сторінка для медитацій містить в собі бібліотеку з файлами (музика, відео), а також таймер з налаштуваннями: вімкнути музику чи відео, вимкнути сповіщення, обрати кількість секунд, хвилин, годин. Для зручної роботи з бібліотекою її поділено на розділи: стрес, сон, концентрація, навчання, відпочинок, пробудження. В кожному розділі додано відповідні до теми матеріали, фактично розділ є фільтром для медитативних файлів.

Для музики та відео на сторінці для медитації розташовано відповідний плеєр, в якому можна перемикати гучність чи застосувати перемотування, також лише для відео є функція «Розгорнути на весь екран».

4.2 Реалізація програмних компонентів бізнес-логіки геймифікованого трекеру звичок

4.2.1 Реєстрація, авторизація та адаптивний інтерфейс

Реєстрація є досить важливою частиною функціоналу сайту. Більшість веброзробників додають можливість реєстрації та авторизації, авторизовані

користувачів зазвичай мають більше прав доступу, в той час, коли не авторизовані користувачі – можуть лише переглядати сторінки без можливості додавати щось або змінювати на сайті. Скоріш з все, це обумовлено тим, що у разі випадку неналежної поведінки користувача (спроби образити інших користувачів, агресія або вживання нецензурних висловів / слів) – можна обмежити можливості конкретного користувача.

Вебзастосунок «HabiTracker» не є виключенням, через що, після авторизації користувача з'являється меню, до функціоналу якого мають доступ лише зареєстровані користувачі. За можливість реєстрації, авторизації та відновлення паролю відповідають наступні файли проєкту:

1) За візуальну частину (оформлення та відображення):
   – login.blade.php;
   – register.blade.php;
   – verify.blade.php;
   – confirm.blade.php;
   – email.blade.php;
   – reset.blade.php.
2) За операції та маніпулювання даними:
   – ForgotPasswordController.php;
   – LoginController.php;
   – RegisterController.php (рис. 4.7);
   – ResetPasswordController.php;
   – VerificationController.php.

```php
protected function create(array $data)
{
    return User::create([
        'name' => $data['name'],
        'email' => $data['email'],
        'password' => Hash::make($data['password']),
    ]);
```

Рисунок 4.7 – Фрагмент коду RegisterController.php

Адаптивність інтерфейсу в першу чергу впливає на вигляд меню користувача [21]. Якщо браузер розгорнуто на весь екран меню матиме наступний вигляд (рис. 4.8). Якщо вікно браузера стиснуто та для звичайного меню користувача не вистачає місця, меню стає компактнішим (рис. 4.9), зберігаючи весь функціонал, для зручного використання додано слайдер.

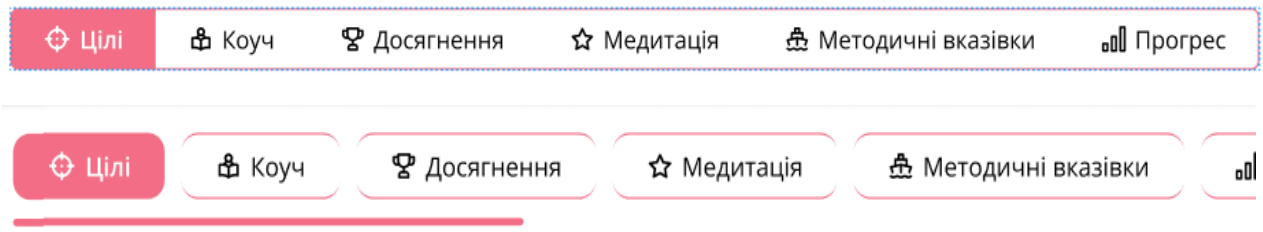


Рисунок 4.8-9 – Вигляд меню користувача повністю та при зменшені вікна

Така зміна відбувається автоматично та не потребує втручань зі сторони користувача, тому є досить зручною, таким чином інформація, що розміщується на сторінці, адаптується під розмір екрана застосунку, тому і має назву адаптивний інтерфейс. Так, наприклад, блок тексту з теоретичною частиною підрозділів також змінює своє розташування в залежності від розміру вікна браузера, меню прийнято закріплювати зверху для зручності, тому воно не змінює своє положення.

4.2.2 Головна сторінка (Home page)

Головна сторінка (рис. 4.10) – обличчя сайту. На ній має бути розміщено:

– коротку інформацію про сайт;

– шапку, в якій, в залежності від стану авторизації, розміщується меню користувача, посилання на реєстраційну форму або форму авторизації, а також ;

– меню, що дозволяє здійснювати перехід на інші сторінки.

За відображення домашньої сторінки відповідає файл home.blade.php (рис. 4.11), а за функції та операції на сторінці – файл HomeController.php (рис. 4.12).

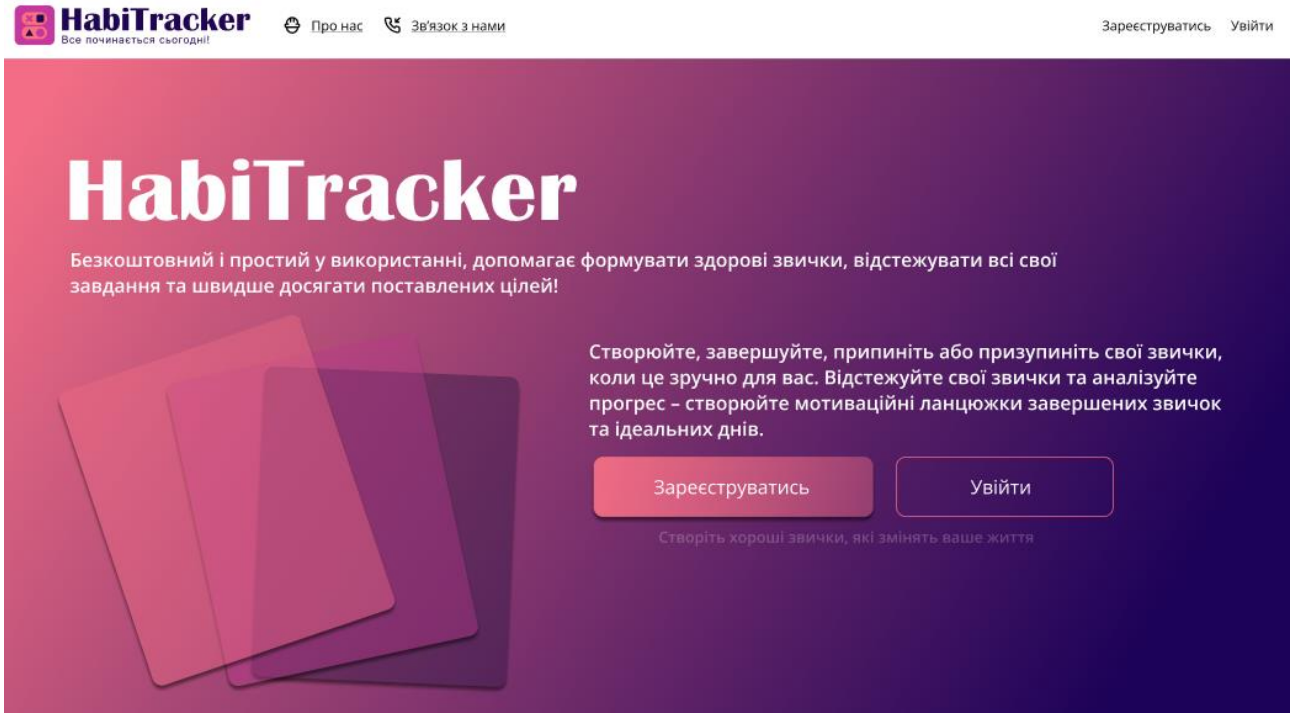


Рисунок 4.10 – Вигляд головної сторінки

```php
<?php
/** @var \App\Services\PageService\Models\Page $page */
?>
@extends("page::layouts.frontend.layout")

@section('content')
    <div class="page page-article">
        <div class="container">
            <div class="article">
                <div class="article-main">
                    <header class="page-header">
                        <a href="/articles" class="back_to_blog">
                            <svg version="1.0" xmlns="http://www.w3.org/2000/svg" width="453.333" height="346.667" viewBox="0 0 340 260">
                                <path d="M211.2 16.5c-5.6 4.7-7.8 13.7-5.1 21.2.5 1.7 15.8 20.1 33.9 40.9l32.8 37.9-129.3.5c-128.9.5-129.4.5-132.2 2.6-4.1 3-6.3 8-6.3 14.4s2.2 11.4 6.3 14.4c2.8 2.1 3.3 2.1 132.2 2.6l129.3.5-32.8 37.9c-18.1 20.8-33.4 39.2-33.9 40.9-2.7 7.5-.5 16.5 5.1 21.2 3.7 3.1 11.8 3.5 16 .7 3.4-2.2 93.9-106.4 95.5-110 1.7-3.7 1.6-12.6-.1-16.5-1.7-3.8-9 0-5.8.5-8.1 2.5z"></path>
                            </svg>
                            {{ trans('general.go_back') }}
                        </a>
                        <h1 class="header-text">{{ $page->title }}</h1>
                    </header>
                    <div class="article-content">
                        @php($image = $page->medias->first()->urlPublic('original'))
                        <div class="page-thumbnail article-image">
                            @if($image)
                                <img src="{{ $image }}" alt="minimum viable product">
                            @endif
                        </div>
                        @foreach($page->blocks as $block)
                            {!! $block->render("page::layouts.frontend.blocks") !!}
                        @endforeach
                    </div>
                </div>
            </div>
        </div>
    </div>

    @include("page::layouts.frontend.blocks-static._default.footer")
@endsection
```

Рисунок 4.11 – Код файлу home.blade.php

```php
<?php

namespace App\Services\PageService\Controllers\Frontend;

use App\Services\PageService\Controllers\Controller;
use App\Services\PageService\PageService;
use Illuminate\Http\Request;

/
 * Class HomeController
 * @package App\Services\PageService\Controllers\Frontend
 */
class HomeController extends Controller
{
    public function home(Request $request)
    {
        / @var PageService $pageService */
        $pageService = app('page');
        $page = $pageService->byKey('home');

        if (!$page) {
            abort(404);
        }

        return $this->viewFrontend('page', compact(
            'page', 'request'
        ));
    }
}
```

Рисунок 4.12 – Код файлу HomeController.php

Таким чином кінцевий вигляд головної сторінки вебзастосунку геймифікованого трекеру звичок «HabiTracker» повністю задовільняє всі вимоги, що описані у першому розділі кваліфікаційної роботи бакалавра.

4.2.2 Додавання звичок і використання медіа для медитації

Основними функціями застосунку є додавання та виконання CRUD-операцій над звичкою користувачем та використання бібліотеки файлів для медитації (авдіо, відео) з можливою фільтрацією та сортуванням файлів. Після виконання операцій та при збереженні даних, звичка зберігається у БД, після чого, як зазначено в патерні MVC [22], сторінка з даними оновлюється та демонструється користувачу (рис. 4.13) вже із внесеними та збереженими змінами.

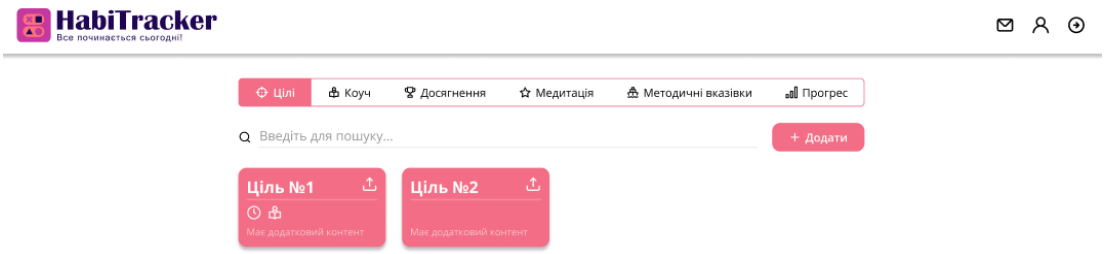


Рисунок 4.13 – Вигляд демонстрації звичок

За виконання дій на сторінці відповідає файл HabitCRUDController.php (рис. 4.14). Контроллер відображає всі необхідні користувачу CRUD-операції, задля чого містить функції з відповідними назвами (create, update тощо).

```php
public function index(Request $request): AnonymousResourceCollection
{
    return ConfigResource::collection(
        Config::query()->paginate(
            $request->input('per_page', 10)
        )
    );
}

public function store(StoreConfigRequest $request): ConfigResource
{
    return new ConfigResource(
        Config::query()->create($request->only([
            'key', 'name', 'value'
        ]))
    );
}
public function show(Config $config): ConfigResource
{
    return new ConfigResource($config);
}

public function update(UpdateConfigRequest $request, Config $config): ConfigResource
{
    return new ConfigResource(
        tap($config)->update($request->only([
            'key', 'name', 'value'
        ]))
    );
```

Рисунок 4.14 – Функції зміни звички

```php
public function store(StoreMediaRequest $request): MediaResource
{
    $this->authorize('store', Media::class);
    $file = $request->file('file');

    try {
        $media = $this->mediaService->uploadSingle(
            (string) $file,
            $file->getClientOriginalName(),
            $request->input('type'),
            $request->input('sync_presets', [
                'thumbnail'
            ]));
    } catch (Exception $exception) {
        log_error(sprintf(
            "Media uploading failed: %s",
            $exception->getMessage()
        ));
    }

    return new MediaResource($media ?? null);
}

public function update(UpdateMediaRequest $request, Media $media): MediaResource
{
    $this->authorize('update', Media::class);

    $translations = $request->input('translations', []);

    foreach ($translations as $locale => $translation) {
        $media->translateOrNew($locale)->fill(array_merge($translation, [
            'media_id' => $media->id
        ]))->save();
    }

    return new MediaResource($media ?? null);
}
public function show(Media $media): MediaResource
{
    $this->authorize('show', $media);

    return new MediaResource($media);
}
```

Рисунок 4.15 – Функції роботи з файлами для медитації

Використання файлів з бібліотеки для медитації предбачає їх програвання та перегляд. На сторінці медитації можна відфільтрувати за типом файлу для зручного пошуку (авдіо чи відео), чи скористатися пошуком за назвою.

4.3 Тестування вебзастосунку

Для отримання якісного та зручного вебзастосунку перед виданням його необхідно протестувати. При наявності проблем або багів необхідно виправити їх до етапу надання доступу користувачам [23]. Дані результатів тестування містять успішні сценарії виконання та розширення, для опису ситуацій з виведенням попереджень про помилку системою, які наведено в таблицях 4.1-4.6.

Таблиця 4.1 – Реєстрація у вебзастосунку

Діючі особи	Гість, система
Мета	Створити обліковий запис.
Передумова	Користувач не авторизований в системі.
Успішний сценарій: 1. Користувач переходить до форми реєстрації. 2. Користувач заповнює необхідні дані (ім’я, прізвище, e-mail, статус, отримання розсилки). 3. Користувач надсилає дані системі. 4. Система оброблює обліковий запис та надсилає лист із паролем для входу та посиланням на підтвердження пошти на вказану електронну адресу. 5. Користувач переходить за посиланням. 6. Система зберігає обліковий запис. 7. Система переводить користувача на головну сторінку.	
Сценарій успішний. Збережено новий обліковий запис.	

Кінець таблиці 4.1

Розширення	
1а	Не всі поля заповнені. Система виводить відповідне повідомлення під кожним незаповненим полем. Результат: користувач не зареєстрований.
2а	Користувач вводить вже існуючі дані. Система виводить відповідне повідомлення. Результат: користувач не зареєстрований.
Усі сценарії розширення успішно виконані.	

Таблиця 4.2 – Авторизація в системі

Діючі особи	Гість, система
Мета	Виконати авторизацію.
Передумова	Користувач не авторизований в системі.
Успішний сценарій: 1. Користувач переходить на сторінку авторизації. 2. Користувач заповнює необхідні дані (пошту, пароль). 3. За бажанням користувач відмічає функцію «Запам’ятати мене». 4. Система валідує дані у БД. 5. Система переводить користувача на головну сторінку.	
Сценарій успішний. Авторизацію виконано	
Розширення	
1а	Не всі поля заповнені. Система виводить відповідне повідомлення під кожним незаповненим полем. Результат: користувач не авторизований.
2а	Не існує такого користувача або невірний пароль. Система виводить відповідне повідомлення. Результат: користувач не авторизований.
Усі сценарії розширення успішно виконані.	

Таблиця 4.3 – Редагування облікового запису

Діючі особи	Користувач, система
Мета	Налаштувати обліковий запис.
Передумова	Користувач авторизованийв в системі.
Успішний сценарій: 1. Користувач переходить до облікового запису. 2. Переходить на сторінку налаштувань завдяки використанню кнопки «Редагувати дані». 3. Виконує необхідні зміни та надсилає відредаговані дані на збереження. 4. Система зберігає зміни в БД та оновлює дані для відображення користувачу.	
Сценарій успішний. Збережено зміни облікового запису.	
Розширення	
1а	Користувач залишив обов’язкові поля порожніми. Система виводить відповідне повідомлення про помилку. Результат: зміни облікового запису не збережено.
Усі сценарії розширення успішно виконані.	

Таблиця 4.4 – Здійснення пошуку матеріалів для медитації з фільтрами

Діючі особи	Користувач, система
Мета	Знайти необхідний файл з бібліотеки для медитації
Передумова	Користувач авторизований.
Успішний сценарій: 1. Користувач переходить до сторінки медитації. 2. Вводить у рядку пошуку назву матеріалу, що бажає знайти. 3. Обирає опції (авдіо або відео) та сортує за ревалентністю матеріалу. 4. Система здійснює пошук відповідно до заданих фільтрів та назви файлу.	
Сценарій успішний. Знайдено необхідний матеріал.	
Розширення	

Кінець таблиці 4.4

1а	За обраними фільтрами та назвою не знайдено матеріал для відображення. Система виводить відповідне повідомлення. Результат: файл з бібліотеки для медитації не знайдено.
Усі сценарії розширення успішно виконані.	

Таблиця 4.5 – Редагування звички

Діючі особи	Користувач, система
Мета	Редагувати звичку.
Передумова	Користувач авторизований.
Успішний сценарій: 1. Користувач переходить до сторінки із звичками. 2. Користувач обирає звичку та натискає кнопку «Редагувати». 3. Виконує необхідні зміни та надсилає відредаговані дані на збереження. 4. Система зберігає дані до БД. 5. Система перенаправляє користувача на сторінку перегляду даних про звичку.	
Сценарій успішний. Обрану звичку відредаговано.	
Розширення	
1а	Не всі поля заповнено. Система виводить відповідне повідомлення. Результат: зміни до даних звички не внесено.
2а	Внесені дані не відповідають необхідному типу даних. Система виводить відповідне повідомлення. Результат: зміни до даних звички не внесено.
Усі сценарії розширення успішно виконані.	

Таблиця 4.6 – Додавання звички

Діючі особи	Користувач, система
Мета	Додати пару до розкладу
Передумова	Користувач авторизований.

Кінець таблиці 4.6

<table>
<tr><td colspan="2">Успішний сценарій:
1. Користувач переходить до сторінки звичок та натискає «Додати звичку».
2. Система відображає вікно з формою для додавання звички.
3. Користувач заповнює необхідні дані та натискає «Зберегти».
4. Система зберігає дані до БД.
5. Система перенаправляє користувача на сторінку перегляду списку звичок з відображенням створенної звички.</td></tr>
<tr><td colspan="2">Сценарій успішний. Додано нову звичку до списку існуючих.</td></tr>
<tr><td colspan="2">Розширення</td></tr>
<tr><td>1а</td><td>Користувач залишив обов’язкові поля порожніми. Система виводить відповідне повідомлення. Результат: створювану звичку не збережено.</td></tr>
<tr><td>2а</td><td>Внесені дані не відповідають необхідному типу даних. Система виводить відповідне повідомлення. Результат: створювану звичку не збережено.</td></tr>
<tr><td colspan="2">Усі сценарії розширення успішно виконані.</td></tr>
</table>

Висновки до розділу 4

В четвертому розділі наведено проведену роботу із кодування front-end та back-end частин вебзастосунку геймифікованого трекеру звичок. За рахунок чого закріплено навички роботи з мовами програмування PHP та JavaScript, їх фреймворками, а також створення та впровадження в систему БД – MySQL.

Проведено тестування функціоналу розробленого вебзастосунку. На основі результатів тестування складено опис основних функцій з можливими успішними сценаріями виконання з розширеннями, для відображення можливих дій користувача та реакцією системи на певні ці дії. Завдяки проведеному аналізу, під час повторного тестування усунено знайдені помилки та модернізовано деякі функції в коді.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи бакалавра автоматизовано процес формування звичок за рахунок розробки програмного забезпечення – вебзастосунку геймифікованого трекеру звичок.

Для досягнення визначеної мети вирішено поставлені завдання:

– проаналізовано предметну область та застосунки-аналоги;

– відокремлено проблеми та знайдено рішення їх усунення;

– визначено функціонал застосунку;

– створено блок-схеми алгоритму роботи геймифікованого трекеру звичок та необхідні UML-діаграми;

– розроблено дизайну досягнень та макету застосунку;

– зібрано та проаналізовано інформацію методичних рекомендацій;

– розроблено front-end частину вебзастосунку на базі технології Vue.js, Bootstrap;

– розроблено front-end частину вебзастосунку на базі технології Laravel, MySQL.

Завдяки проведеному дослідженню предметної області визначено актуальність автоматизації процесу формування звички за рахунок розробки геймифікованого трекеру звичок «HabiTracker» та сферу застосування застосунку. За результатами аналізу застосунків-аналогів та визначенню їх переваг та недоліків сформовано вимоги до ПЗ, що розроблялося. В результаті сформовано специфікацію вимог до геймифікованого трекеру звичок та технічне завдання.

Для моделювання та проєктування системи розроблено 7 видів UML-діаграм, задля найбільш зрозумілої демонстрації роботи та структури застосунку. Відповідно до технічного завдання обрано найбільш придатні для розробки вебзастосунку технології, розроблено front-end та back-end частини ПЗ. Проведено тестування застосунку та усунено знайдені помилки. Результатом проведеної роботи є вебзастосунок геймифікований трекер звичок.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1) Help. Balance: Personalized Meditation and Sleep Guidance. URL: https://www.balanceapp.com/balance-terms (дата звернення: 01.06.2022).

2) Amazing habit tracking features. HabitShare - A Social Habit Tracker to Keep You Accountable for iOS & Android. URL: https://habitshareapp.com/#features (дата звернення: 01.06.2022).

3) Розенвассер Д. М., Трикуліч П. П. Розробка веб-застосунку–«здоров'я». Trends and prospects development of science and practice in modern environment : The X International Science Conference, Geneva, 22 листопада 2021. с. 383.

4) The research of PHP development framework based on MVC pattern / W. Cui та ін. 2009 fourth international conference on computer sciences and convergence information technology, м. Seoul, Korea, 24–26 листоп. 2009 р. 2009. URL: https://doi.org/10.1109/iccit.2009.130 (дата звернення: 01.06.2022).

5) The mvc-web design pattern. 7th international conference on web information systems and technologies, м. Noordwijkerhout, Netherlands, 6–9 трав. 2011 р. 2011. URL: https://doi.org/10.5220/0003296901270130 (дата звернення: 01.06.2022).

6) Turan M. Integrating software metrics with UML class diagrams. Lecture notes on software engineering. 2015. Т. 3, № 3. С. 220–224. URL: https://doi.org/10.7763/lnse.2015.v3.194 (дата звернення: 01.06.2022).

7) More P., Phalnikar R. Generating UML diagrams from natural language specifications. International journal of applied information systems. 2012. Т. 1, № 8. С. 19–23. URL: https://doi.org/10.5120/ijais12-450222 (дата звернення: 01.06.2022).

8) Test case generation from UML-diagrams using genetic algorithm / R. Kumar Sahoo та ін. Computers, materials & continua. 2021. Т. 67, № 2. С. 2321–2336. URL: https://doi.org/10.32604/cmc.2021.013014 (дата звернення: 01.06.2022).

9) Gnesi S., Latella D., Massink M. Model checking UML Statechart diagrams using JACK. HASE 99: fourth IEEE international symposium on high assurance systems engineering, м. Washington, DC, USA. URL: https://doi.org/10.1109/hase.1999.809474 (дата звернення: 01.06.2022).

10) Mokarat C., Vatanawood W. UML component diagram to acme compiler. 2013 international conference on information science and applications (ICISA), м. Suwon, Korea (South), 24–26 черв. 2013 р. 2013. URL: https://doi.org/10.1109/icisa.2013.6579469 (дата звернення: 01.06.2022).

11) Package diagram. Monographs in computer science. New York, NY. С. 133–154. URL: https://doi.org/10.1007/0-387-23803-4_7 (дата звернення: 01.06.2022).

12) Academy C. P. Php: the ultimate crash course to learn PHP with practical computer coding exercises. Independently Published, 2019. 227 с.

13) Prokofyeva N., Boltunova V. Analysis and practical application of PHP frameworks in development of web information systems. Procedia computer science. 2017. Т. 104. С. 51–56. URL: https://doi.org/10.1016/j.procs.2017.01.059 (дата звернення: 01.06.2022).

14) Kromann F. M. Web services. Beginning PHP and MySQL. Berkeley, CA, 2018. С. 443–460. URL: https://doi.org/10.1007/978-1-4302-6044-8_18 (дата звернення: 01.06.2022).

15) Google trends: Laravel, CodeIgniter, Symfony, Yii, CakePHP. Google Trends. URL: https://trends.google.com/trends/explore?q=laravel, CodeIgniter,Symfony,Yii,CakePHP&hl=en-US (дата звернення: 01.06.2022).

16) Dikta G., Scheer M. Introduction. Bootstrap methods. Cham, 2021. P. 1–7. URL: https://doi.org/10.1007/978-3-030-73480-0_1 (дата звернення: 01.06.2022).

17) Google trends: MySQL, Oracle. Google Trends. URL: https://trends.google.com/trends/explore?q=MySQL,Oracle%20DB&hl=en-US (дата звернення: 01.06.2022).

18) Yadav N., Rajpoot D. S., Dhakad S. K. LARAVEL: A PHP framework for e-commerce website. 2019 fifth international conference on image information processing (ICIIP), м. Shimla, India, 15–17 листоп. 2019 р. 2019. URL: https://doi.org/10.1109/iciip47207.2019.8985771 (дата звернення: 01.06.2022).

19) Li D., Chen Y. Data migration research between relational database and nonrelational database. Multimedia, communication and computing application. 2015. С. 77–81. URL: https://doi.org/10.1201/b18512-19 (дата звернення: 01.06.2022).

20) Meier A., Kaufmann M. NoSQL databases. SQL & NoSQL Databases. Wiesbaden, 2019. С. 201–218. URL: https://doi.org/10.1007/978-3-658-24549-8_7 (дата звернення: 01.06.2022).

21) Adapting user interfaces with model-based reinforcement learning / K. Todi та ін. CHI '21: CHI conference on human factors in computing systems, м. Yokohama Japan. New York, NY, USA, 2021. URL: https://doi.org/10.1145/3411764.3445497 (дата звернення: 01.06.2022).

22) AutoCRUD: Automatic generation of CRUD specifications in interaction flow modelling language / R. Rodriguez-Echeverria та ін. Science of computer programming. 2018. Т. 168. С. 165–168. URL: https://doi.org/10.1016/j.scico.2018.09.004 (дата звернення: 01.06.2022).

23) Malevris N. Software testing. Journal of computer engineering & information technology. 2013. Vol. 02, no. 01. URL: https://doi.org/10.4172/2324-9307.1000e105 (дата звернення: 01.06.2022).

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

ГЕЙМИФІКОВАНИЙ ТРЕКЕР ЗВИЧОК

СПЕЦІАЛЬНА ЧАСТИНА З ОХОРОНИ ПРАЦІ

ОХОРОНА ПРАЦІ НА РОБОЧИХ МІСЦЯХ ФАХІВЦІВ З ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Спеціальність 121 «Інженерія програмного забезпечення»

121 – КРБ.1 – 409.21810904

Студентка _____ С. В. Бондаренко
підпис
«__» _____ 2022 р.

Консультант канд. техн. наук, доцент _____ А. О. Алексєєва
підпис
«__» _____ 2022 р.

Миколаїв – 2022

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ 3
ВСТУП 4
1 ОЦІНКА УМОВ ПРАЦІ ФАХІВЦІВ З ІТ-ТЕХНОЛОГІЙ У РОБОЧОМУ ПРИМІЩЕННІ 5
1.1 Опис виробничого приміщення. Характеристики та параметри, план, обладнання та умови праці 5
1.2 Проведення розрахунків для визначення рівню природнього освітлення у робочій кімнаті 8
2 ПОРЯДОК ДІЙ ПЕРСОНАЛУ У НАДЗВИЧАЙНІЙ СИТУАЦІЇ, ЩО ПОВ'ЯЗАНА З ПОЖЕЖНОЮ НЕБЕЗПЕКОЮ 13
2.1 Алгоритм виконання дій персоналом при виникненні пожежі 13
2.2 Розробка та опис плану евакуації з підприємства на якому виконувався процес розобки 15
ВИСНОВКИ 17
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ 18
ДОДАТОК А Дані параметрів робочого приміщення, їх позначення та одиниці вимірювання 19
ДОДАТОК Б План евакуації на підприємстві з позначеннями 20

ПЕРЕЛІК СКОРОЧЕНЬ

ООП – основи охорони праці

БНіП – будівельні норми і правила

ДБЖ – джерело безперебійного живлення

ЗІЗ – засоби індивідуального захисту

КПО – коефіцієнт природного освітлення

КРБ – кваліфікаційна робота бакалавра

ОП – охорона праці

ПВХ – полівінілхлорид

ПЗ – програмне забезпечення

ПСК – пояс світлового клімату

ІТ – information technology

ВСТУП

На сьогодні, актуальним та головним обов'язком роботодавця є забезпечення комфортних умов для праці з врахуванням норм та правил безпеки. Оскільки середовище усіма чинниками має вплив на здоров'я та самопочуття людини, яка в ньому працює. Окрім того, дані фактори мають змогу впливати як і на продуктивність роботи, так і в цілому на працездатність робітника.

Основна мета системи ОП полягає в зменшенні небезпечних та шкідливих для робітника факторів, що мають змогу впливати на роботу працівників, у виробничому середовищі, за допомогою спеціально визначених норм. ОП базується на науках: безпека життєдіяльності, екологія, ергономіка, тощо.

Обов'язком кожного інженера є наявність відведеного обсягу знань у галузі ОП, оскільки це важливий інструмент при вирішені задач, умовою яких є створення безпечних умов праці у підприємстві. У свою чергу, наявний досвід від підприємств демонструє, що впровадження і використання поодиноких заходів з ОП, надають ефект набагато менший. Тому варто використовувати системний підхід, де усі правила ОП на підприємстві мають використовуються комплексно, та взаємопов'язано [1]. Безпека праці на виробництві має 3 основні складові частини: безпека виробничого обладнання; безпека технологічних процесів; організація та впровадження безпечного алгоритму виконання роботи.

Так у першому розділі представлено аналіз виробничого приміщення: опис приміщення, характеристик та параметрів, обладнання та умови для праці. Крім того, представлено проведений розрахунок природного освітлення у робочій кімнаті та виявлено потреби у додатковому штучному освітленні в темний час доби.

Другий розділ, в свою чергу, демонструє необхідні заходи для забезпечення безпеки робочого процесу та евакуації усіх працівників, зумовленою порушенням правил пожежної безпеки, у разі виникнення на підприємстві пожежної небезпеки. В тому числі інструкцію та порядок дій персоналу та план евакуації з підприємства.

1 ОЦІНКА УМОВ ПРАЦІ ФАХІВЦІВ З ІТ-ТЕХНОЛОГІЙ У РОБОЧОМУ ПРИМІЩЕННІ

1.1 Опис виробничого приміщення. Характеристики та параметри, план, обладнання та умови праці

Кожне з обладнання, що задіяно під час роботи, має відповідати усім вимогам безпеки з початку та до кінця використання, тобто протягом всього часу експлуатації. Стосовно матеріалів робочого обладнання, вони мають бути безпечними під час використання, а саме не бути чинником небезпечної і шкідливої дії токсинів на організм робітників, водночас, навантаження, що виникають в процесі роботи обладнання, в окремих його елементах обов’язково не мають досягати небезпечних значень. У разі неможливості виконання заданих вище умов, є необхідність використання ЗІЗ: окулярів, огороджень, спеціальних рукавиць тощо.

Розташування робочої кімнати – на першому поверсі, 2-х поверхового будинку. У представленій кімнаті проходив весь процес роботи із розробки ПЗ, геймифікованого трекеру звичок, від проведення аналізу та проєктування застосунку до кодування та тестування ПЗ. Для опису більш детального аналізу приміщення, у якому проходив робочій процес, розроблено план приміщення (рис. 1.1), з використанням середовища Visio 2019.

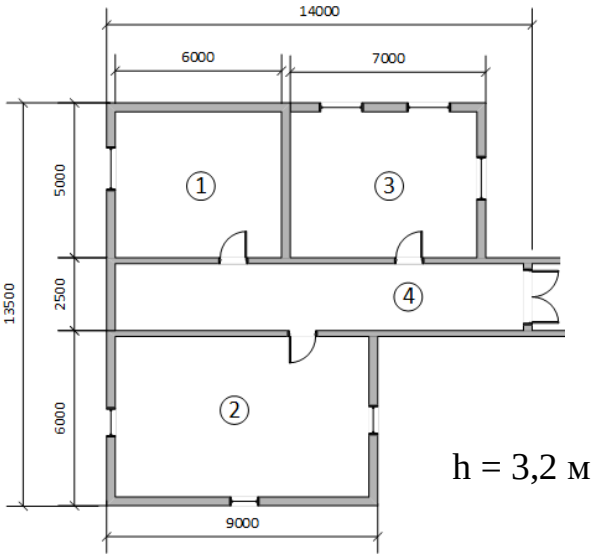


Рисунок 1.1 – План робочого приміщення

Весь робочій процес проходив у одній із робочих кімнат даного підприємства (кімната №1).

Основні параметри кімнати наступні:

– довжина кімнати – 9 м;
– ширина кімнати – 6 м;
– стіни кімнати – гладкі, бетонні з 3-а вікнами та 1-и розпашними всередину дверями;
– висота стіни – 3,2 м;
– товщина стін – 0,2 м;
– стеля – гладка, темна, бетона;
– підлога вкрита лінолеумом світлого тону.

У приміщенні освітлення одностороннє природне, бокове. Крім того у наявності 3 вікна з розмірами c×d = 1,4×1,9 (м), на 2,2 м від робочої поверхні розташована висота їх верхнього краю. Характеристикою вікон є дерев'яні рамами та подвійне листове віконне скло. На відстані приблизно 5 метрів до зовнішньої стінки розташовується робоча поверхня працівника.

Приміщення пристосоване для роботи 2 людей (працівників). Крім того у наявності 1 комп'ютер та 1 ноутбук, які розташовані на 2 столах – на кожному із столів по 1-у пристрою. Навпроти вікна розташоване робоче місце працівника-автора КРБ. Характеристика комп'ютерних столів, що є робочим місцем:

– ширина – 1 м;
– довжина – 0,5 м;
– висота – 0,8 м;
– зовнішній вигляд та матеріали: світлого кольору, дерев'яний.

Кожне з робочих місць обладнане:

– 1 столом;
– 2-а стільцями;
– 2 монітори;
– системний блок *(лише для комп'ютера)*;
– 1-а пара навушників;

– периферійним обладнанням;
– ДБЖ.

Крім того, задля отримання більшої кількості освітлення робоче місце розміщено біля вікна, тобто знаходиться під природним освітленням. Через що проведено розрахунки на природне освітлення задля перевірки рівня освітлення у кімнаті із робочим місцем робітника. Оскільки при розрахунках важливо враховувати розташування робочого місця у кімнаті (рис. 1.2), а також розташування протилежного будинку (рис. 1.3) – їх розміщення зображено графічно за допомогою використання середовища Visio 2019.

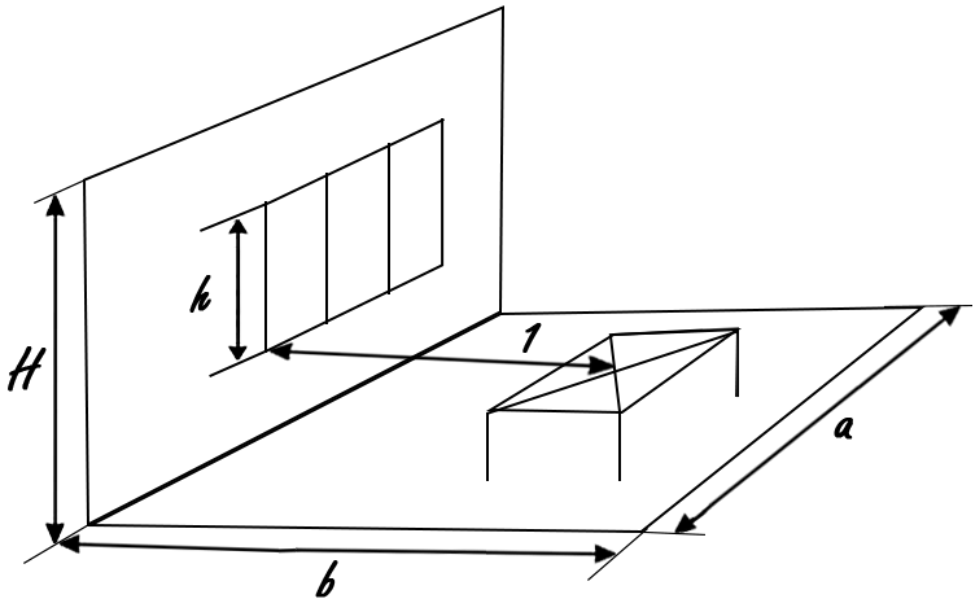


Рисунок 1.2 – Розташування робочої поверхні у кімнаті

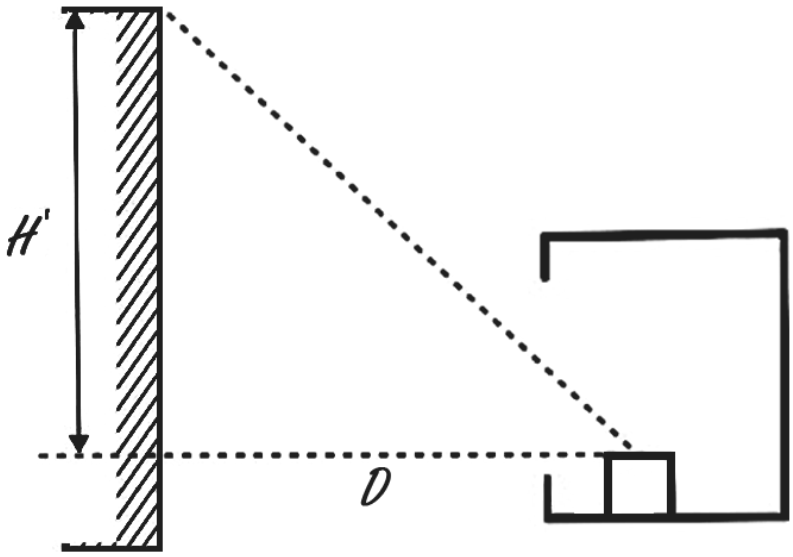


Рисунок 1.3 – Розташування протилежної будівлі

Конкретні розрахунки потребують значень розмірів приміщення (висота, ширина тощо), а також важливою є інформація, що стосується будівлі, такі як відстань від робочої поверхні до зовнішньої стіни та інше – всі дані наведено у Додатку А. Розташування будівлі, в якій розміщено виробниче приміщення, в м. Миколаїв, Миколаївської області, що належить до IV поясу світлового клімату (ПСК), коефіцієнт якого m= 0,9, і південніше 50° північної широти [4].

1.2 Проведення розрахунків для визначення рівню природнього освітлення у робочій кімнаті

Поняття *природнього освітлення* базується на сонячному промінні та розсіяному світлому небосхилу, тобто кількість світла, що потрапляє через світлові прорізи (вікна) в зовнішніх огороджувальних конструкціях до приміщення. Вони проникають у виробниче приміщення через світлові прорізи в зовнішніх відгороджуючих конструкціях [5].

При створенні природного освітлення досить важливо врахувати тип та розміщення сумарної площі вікон, оскільки це основний фактор забезпечення необхідного світлового режиму, який в жодному разі не може бути прибраним з розрахунків. Після отримання даних параметрів, наступним етапом є безпосередньо розрахунки природного освітлення, які виконуються за визначеною послідовністю:

1. *Виконання розрахунку нормативного значення КПО відповідно до будівельних норм і правил, для III ПСК –* $e_{\text{н}}^{III}$, *%.*

Отримання початкових значень для розрахунків виконується відповідно до БНіП II - 4 - 79 [3, табл. 3.1]. Відстань стосовно зорових робіт середньої точності, при найменшому розміщенні об’єкта розпізнавання, становить від 1 до 5 мм. Розрахунки при боковому освітленні, враховуючи V – розряд зорової роботи:

$$e_{\text{н}}^{III} = 1{,}0\%.$$

2. *Виконання розрахунку для визначення коефіцієнта світлового клімату для IV ПСК.*

Оскільки приміщення розташовано в м. Миколаїв, Миколаївської області, що належить до IV ПСК, розрахунки проведено для даного ПСК [3]:

$$m = 0{,}9.$$

3. *Виконання розрахунку для визначення коефіцієнта сонячності клімату.*

Розрахунки виконуються згідно з нормами для вікон в зовнішніх стінах будівель, що розташовані у IV ПСК та зорієнтованих по азимуту в діапазоні з 136 до 225 градусів відповідно до рекомендацій [3, табл. 3.3]:

$$c = 0{,}75.$$

4. *Виконання розрахунку для визначення нормованого значення КПО для умов згідно до м. Миколаєва,* $e_н$, *%:*

$$e_н = e_н^{III} \times m \times c = 1 \times 0{,}9 \times 0{,}75 = 0{,}68\%$$

5. *Визначення коефіцієнта запасу, що використовується при розрахунку природного освітлення –* $К_з$.

За рекомендаціями [3]: $К_з = 1{,}3 \ldots 1{,}5$. Прийнято значення $К_з = 1{,}4$.

6. *Визначення відношення довжини приміщення до його ширини, відношення a до b відповідно:*

$$\frac{a}{b} = \frac{9}{6{,}0} = 1{,}5.$$

7. *Визначення відношення ширини приміщення (b) до відстані від верхньої кромки вікна до робочої поверхні, відношення b до h відповідно:*

$$\frac{b}{h} = \frac{6{,}0}{2{,}2} = 2{,}7.$$

8. *Розрахунки для визначення світлової характеристики вікна –* $ŋ_в$.

Значення визначається відповідно до [3, табл. 3.4]:

$$ŋ_в = f \times (\frac{a}{b}, \frac{b}{h}), \text{ при } \frac{a}{b} = 1{,}5 \text{ та } \frac{b}{h} = 2{,}7;$$

$$ŋ_в = 15.$$

9. *Розрахунки для визначення коефіцієнта світлопропускання (прозорість) матеріалу –* τ_1.

Значення визначається відповідно до [3, табл. 3.6] для характеристики «Подвійне листове віконне скло»:

$$\tau_1 = 0{,}8.$$

10. *Визначення коефіцієнта, що враховує втрати світла у віконній рамі.*

Визначається відповідно до [3] для характеристики «Спарені дерев'яні рами»:

$$\tau_2 = 0{,}7.$$

11. *Визначення коефіцієнта, що враховує втрати світла в несучих конструкціях.*

При боковому освітленні відповідно до [3]:

$$\tau_3 = 1.$$

12. *Визначення коефіцієнта, що враховує втрати світла у сонцезахисних пристроях.*

Визначається відповідно до [3] для характеристики «Регульовані внутрішні жалюзі»:

$$\tau_4 = 1.$$

13. *Визначення загального коефіцієнта світло пропускання –* $\tau_{заг}$*:*

$$\tau_{заг} = \tau_1 \times \tau_2 \times \tau_3 \times \tau_4 = 0{,}8 \times 0{,}7 \times 1 \times 1 = 0{,}56.$$

14. *Коефіцієнт відбиття внутрішніх поверхонь приміщення –* ρ*: стелі, стін та підлоги.*

Значення визначаються відповідно до [3]:

– для стелі, відповідно для характеристики «Темна дерев'яна (нефарбована)», $\rho_{\text{стелі}}$ = 27%;

– для стін, відповідно для характеристики «Бетонні з вікнами», $\rho_{\text{стін}}$ = 30%;

– для підлоги, відповідно для характеристики «Лінолеум світлого тону», $\rho_{\text{підлоги}}$ = 32%.

15. *Обчислення площі внутрішніх поверхонь приміщення:* $S_{\text{стелі}}$, $S_{\text{стін}}$, $S_{\text{підлоги}}$, м^2:

$$S_{\text{стелі}} = a \times b = 9 \times 6{,}0 = 54 \text{ м}^2;$$

$$S_{\text{стін}} = 2 \times (a+b) \times H = 2 \times (9+6{,}0) \times 3{,}2 = 96 \text{ м}^2;$$

$$S_{\text{підлоги}} = a \times b = 9 \times 6{,}0 = 54 \text{ м}^2.$$

16. *Обчислення середнього значення коефіцієнта відбиття внутрішніх поверхонь приміщення:*

$$\rho_{\text{сер.}} = \frac{(\rho_{\text{стелі}} \times S_{\text{стелі}}) + (\rho_{\text{стін}} \times S_{\text{стін}}) + (\rho_{\text{підлоги}} \times S_{\text{підлоги}})}{(S_{\text{стелі}} + S_{\text{стін}} + S_{\text{підлоги}}) \times 100} = \frac{(27 \times 54) + (30 \times 96) + (32 \times 54)}{(54+96+54) \times 100} =$$

$$= \frac{6066}{20400} = 0{,}29.$$

17. *Визначення геометричного співвідношень, таких як a до b; b до h; 1 до b*:

$$\frac{a}{b} = \frac{9}{6} = 1{,}5;$$

$$\frac{b}{h} = \frac{6}{2.2} = 2{,}72;$$

$$\frac{l}{b} = \frac{5}{6} = 0{,}83.$$

18. *Розрахунок коефіцієнта, який враховує підсилення природного освітлення за рахунок відбиття світлового потоку приміщення.*

Відповідно до [3]:

$r_1 = f\ (\rho_{\text{сер.}} \times \frac{a}{b} \times \frac{b}{h} \times \frac{l}{b})$, при $\rho_{\text{сер.}}$ = 0,29, $\frac{a}{b}$ = 1,5, $\frac{b}{h}$ = 2,72, $\frac{l}{b}$ = 0,83; r_1 = 1,5.

19. *Визначення відношення відстані до протилежної будівлі D до висоти карнизу протилежної будівлі над підвіконням H':*

$$\frac{D}{H'} = \frac{23}{12,5} = 1,84.$$

20. *Розрахунок коефіцієнта, який враховує вплив протилежної будівлі на освітленість у приміщенні,* $К_{буд.}$.

Відповідно з [3] при $\frac{D}{H'} = 1,84$; $К_{буд.} = 1,1$.

21. *Обчислення площі вікон, яка є необхідною для нормованого коефіцієнта освітлення природним освітленням –* $S_{в.}$, $м^2$*:*

$$S_{в.} = \frac{e_н \times К_{з.} \times \eta_{в.} \times S_{підлоги} \times К_{буд.}}{t_{заг.} \times r_1 \times 100} = \frac{0,68 \times 1,4 \times 15 \times 54 \times 1,1}{0,56 \times 1,5 \times 100} = 10,09 \text{ м}^2.$$

У робочому приміщенні розташовано 3 вікна, їх розміри – d×c = 1,4×1,9. Останнім етапом розрахунків є визначення дійсної площі вікон в робочому приміщенні ($S_{факт}$), після чого можна визначити актуальний рівень природної освітленості розрахунковому значенню:

$$S_{факт} = n_в \times c \times d = 3 \times 1,4 \times 1,9 = 7,98 \text{ м}^2.$$

При порівнянні значень, отриманих при розрахунках, тобто дійсного значення з розрахунковим, можна визначити, що для комфортної та безпечної роботи у виробничому приміщенні необхідно встановити додаткове штучне освітлення, оскільки наявне значення рівня природного освітлення є недостатнім. Важливо враховувати, що при виконанні робіт в приміщенні варто використовувати системи комбінованого освітлення [6].

Крім того, для забезпечення комфортніших умов роботи у темний час доби, варто забезпечити приміщення розрядними лампами для комфортного освітлення. Оскільки це необхідність для здоров'я та безпеки працівника, враховуючи особливість роботи фахівців з інформаційних технологій, які тривалий час працюють за екранами моніторів.

2 ПОРЯДОК ДІЙ ПЕРСОНАЛУ У НАДЗВИЧАЙНІЙ СИТУАЦІЇ, ЩО ПОВ'ЯЗАНА З ПОЖЕЖНОЮ НЕБЕЗПЕКОЮ

2.1 Алгоритм виконання дій персоналом при виникненні пожежі

Оскільки кожне з робочих місць фахівців з інформаційних технологій обладнане електронними пристроями, дротами та мікросхемами, є досить високий ризик виникнення пожежної небезпеки. Оскільки навіть при наявності в пристрої плавкого запобіжника не унеможливлюється перенавантаження системи, яке спричинить вихід з ладу вбудованих мікросхем, які в свою чергу можуть стати осередком пожежі. Також не треба виключати можливість пробиття ізоляції дротів та кабелів через підвищену напругу, особливо в місцях механічних пошкоджень від зносу матеріалів ізоляції за віком або умов використання (підвищена вологість чи перегрівання струмовим навантаженням), що також є досить поширеною причиною пожеж. Тому, окрім дотримання правил безпеки під час робочого процесу та виконання рекомендацій стосовно експлуатації пристроїв, важливим є знання та розуміння організації дій у разі виникнення пожежі.

За умовою, що пожежа трапилася у приміщенні з присутністю працівника, безпосередньо працівнику варто дотримуватися наступних правил [3]:

а) в першу чергу, без зволікань зателефонувати до пожежної служби, за номером 101. Під час дзвінка важливо правильно повідомити інформацію стосовно пожежі: назвати чітко та виразно адресу, кількість поверхів у будівлі, приблизну кількість людей, що можуть знаходитись усередині, а також коли саме почалась пожежа та за яких умов (якщо така інформація є доступною);

б) за можливістю, під час власної евакуації, забрати з собою важливі документи компанії та особисті;

в) якщо пожежа виникла на іншому поверсі – сповістити керівника або відповідального за пожежну безпеку;

г) у разі наявності хмари диму – лягти на підлогу та повзти до дверей для виходу з приміщення;

д) перед відкриттям дверей варто їх перевірити, якщо двері гарячі – то не варто їх відкривати та чіпати взагалі, оскільки полум’я не дозволить вийти, якщо двері палають – в жодному разі не намагатися їх вибити власноруч, оскільки є небезпека займання одягу;

е) за можливістю просуватися до вікна, важливо рухатися до вікна з якого іде повітря всередину приміщення;

ж) голосно кликати на допомогу;

з) якщо вдається вийти через двері, необхідно за собою обов’язково їх зачиняти [2].

До зобов’язаностей посадової особи, у разі виникнення пожежі на об’єкті, входять наступні дії:

а) перевірка виклику пожежної служби, якщо виклик не було здійснено – повідомити власноруч, після чого зв’язатися з власником підприємства та повідомити про обставини та поточну ситуацію на підприємстві;

б) у разі наявності загрози життю, необхідно без зволікань розпочати евакуацію працівників підприємства, після чого переконатися в наявності усіх евакуйованих;

в) перевірка стану евакуйованих, за необхідністю виклик швидкої медичної допомоги, за номером 103;

г) зупинка всіх можливих процесів роботи на підприємстві;

д) здійснннення вимкнення електроенергії в разі необхідності, за винятком вимкнення систем протипожежного захисту, після чого зупинення транспортуючих пристроїв, за наявності;

е) перевірка та увімкнення оповіщення чи сирени, що сигналізує про пожежну загрозу на підприємстві, якщо це не було зроблено раніше;

ж) очікування рятувальної групи, під час чого організування рятувальній групі безперешкодного під’їзду до підприємства та забезпечення їх додатковим джерелом речовин для гасіння, за наявністю;

з) при наявності можливості, організація захисту цінних речей компанії;

и) контроль стосовно дотримання правил техніки безпеки працівниками, що беруть участь у гасінні пожежі чи евакуації інших працівників [2].

2.2 Розробка та опис плану евакуації з підприємства на якому виконувався процес розобки

Оскільки при пожежі є висока ймовірність випаровування та горіння токсичних речовин, необхідно без зволікань починати евакуацію, тому що вплив токсичних речовин несе в собі загрозу для здоров'я та життя працівника. Для запобігання паніки та для допомоги в орієнтуванні працівникам, на випадок пожежної небезпеки, має бути розміщено план евакуації з приміщення з такими маркуваннями як: розташування евакуаційний вихід (або виходи), вогнегасників, телефонного апарату, а також вказано напрямок руху при евакуації стрілками. Крім того важливо вказати на плані знак просторової орієнтації «Ви перебуваєте тут», для розуміння плану працівником.

Щоб визначити чи можна вихід (чи виходи) вважати евакуаційними необхідно звернути увагу на відповідні критерії, якщо вихід відповідає хоч одному з них – вихід вважається евакуаційним:

− є провідним на шляху евакуації: безпосередньо назовні чи в безпечну зону;

− є призначеним для евакуації людей при надзвичайних ситуаціях, зазвичай такі виходи містять за замовчуванням маркування «Основний (або запасний) евакуаційний вихід» або «Аварійний вихід»;

− є провідним на шляху евакуації, призначений для екстреного та швидкого виходу з будівлі, на випадок надзвичайних ситуацій. Зазвичай такі виходи розраховані на евакуацію великої кількості людей, які зазвичай не ознайомлені взагалі або частково з розташуванням виходів, їх технічною оснащеністю або для дезорієнтованих, через паніку, людей [9].

Відповідно до загальновстановлених правил та норм, план евакуації має бути розміщених у просторих та доступних місцях з гарним освітленням, задля того, щоб кожен із співробітників мав змогу з ним ознайомитися. Як правило,

для розміщення обирають коридори та приміщення, що розташовані біля основних виходів з будівлі. План евакуації на підприємстві, де проходив процес розробки застосунку геймифікованого трекеру звичок, а також умовні позначення продемонстровано у Додатку Б.

План розроблено для розташування у кімнаті №2, де безпосередньо відбувався робочій процес, тому має відповідний знак просторової орієнтації «Ви перебуваєте тут». Приміщення пристосоване під роботу двох працівників, будівля має один евакуаційний вихід. Кількість вогнегасників розрахована під кількість кімнат, таким чином загальна кількість дорівнює 4-м вогнегасникам. У коридорі (кімната №4) також присутній телефонний апарат для виклику пожежної служби. Евакуаційний вихід зображено за допомогою відповідної позначки, крім того продубльовано з зовнішньої сторони для кращого розуміння плану працівником.

Враховуючи, що двері евакуаційних виходів є двійними, а кількість працівників не перевищує 2 особи, можна зробити висновок, що заторів, безпосередньо при евакуації, не має траплятися. Також варто розробити план дій у разі, якщо осередок пожежі сконцентрований біля виходу з кімнати, що блокує його можливість використання. В такому випадку можна зробити спроби евакуації, скориставшись вікном, це дозволяє відсутність гратів на вікнах.

Важливо, щоб розташування плану евакуації для ознайомлення і вивчення, супроводжували встановлені таблички з інструкцією по виклику пожежної служби. Крім того, оскільки в приміщеннях присутні вогнегасники, всі працівники мають бути ознайомлені з правилами поведінки та роботи з вогнегасниками у разі виникнення пожежної небезпеки. Якщо на підприємстві окрім стандартних переносних вогнегасників розміщено вогнегасники інших видів, це також має супроводжуватися інструктажем з правилами роботи персоналу, а також таблички з інструкціями для нагадування.

ВИСНОВКИ

В спеціальній частині кваліфікаційної роботи бакалавра розглянуто та проаналізовано ситуації, які можуть становити загрозу життю працівника на підприємстві, а також створено алгоритм дій, у разі виникнення загрози життю працівника. Знання інструкцій та правил поведінки персоналом є основною складовою безпеки працівників на підприємстві. Враховуючи фактори ризику на робочих місцях фахівців з інформаційних технологій приділено увагу безпечній роботі з апаратними пристроями: комп'ютери, блоки живлення, дроти тощо.

Крім того, під час виконання роботи проаналізовано та виявлено основні фактори та характеристики виробничого середовища на робочих місцях. Як результат, всі висвітлені в спеціальній частині фактори задовольняють санітарним вимогам. Виявлено проблему в недостатньому рівні природного освітлення, яку вирішено завдяки обладнанню приміщення додатковим штучним освітленням, а саме розрядними лампами. Після усунення проблеми можна вважати, що робоче місце є комфортним та безпечним для працівника, тобто повністю готове для виконання роботи. При виконанні спеціальної частини проведено розрахунки для визначення значення природного освітлення, відповідно до характеристик кімнати, у якій виконувалася робота над розробкою геймифікованого трекеру звичок. Також визначено реальне значення природного освітлення у кімнаті через вікна, відповідно до отриманого значення зроблено висновки, що природне освітлення повністю задовольняє поставлені вимоги.

Описано можливі фактори та ризики виникнення пожежної небезпеки на підприємстві. Оскільки запобігти наслідків для здоров'я та життя персоналу при пожежі можуть інструкції та чіткий алгоритм дій, розроблено план евакуації з відповідними позначками для приміщення, у якому проходила розробка ПЗ, а також сформульовано план дій, правила евакуації персоналу та основні вимоги до швидкої евакуації з приміщення. За результатами виконання спеціальної частини з охорони праці можна зробити висновок, що приміщення відповідає нормам для запобігання утворенню пожежної небезпеки.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Навчальний посібник / Д.В. Зеркалов; за ред. Київ: Основа, 2011.551 с.
2. Київська державна адміністрація. Рекомендації у виникненні пожежі [Електронний ресурс] URL : https://old.kyivcity.gov.ua/content/pamyatka-shchodo-diy-u-razi-vynyknennya-pozhezhi.html (дата звернення: 24.05.2022).
3. Посібник к БНіП II-4-79 «Допомога з розрахунку та проектування природного, штучного та поєднаного освітлення» [Електронний ресурс]. URL : http://tehlit.ru/1lib_norma_doc/6/6141/ (дата звернення: 24.05.2022).
4. Охорона праці в галузі / П. Атаманчук, В. Мендерецький, О. Панчук, Р. Білик. Центр навчальної літератури, 2017. 322 с.
5. Охорона праці в галузі. Академія on-line. URL: http://moodle.socosvita.kiev.ua/moodledata/filedir/c0/02/c002da13808d576e9caa6929ff71663349a9dc6c (дата звернення: 24.05.2022).
6. Штучне освітлення, нормування. [Електронний ресурс]. URL: https://cpo.stu.cn.ua/Oksana/posibnik/770.html#:~:text=2.5–28–2006,%20згідно,з%20фоном%20і%20характеристики%20фону (дата звернення: 24.05.2022).
7. Системи природного та штучного освітлення підприємств галузі. Модульне середовище для навчання. URL: https://msn.khnu.km.ua/pluginfile.php/297860/mod_resource/content/2/Лекція_8_1_Освітлення_нова.pdf#:~:text=бути%20трьох%20видів:-,1.,2 (дата звернення: 24.05.2022).
8. Робоче місце програміста розділ диплому з БЖД. Реферати, курсові, дисертації, дипломи. URL: https://ua-referat.com/Робоче_місце_програміста_розділ_диплому_з_БЖД (дата звернення: 24.05.2022).
9. Практикум з безпеки життєдіяльності. Навчальний посібник./І. Пістун, В. Кочубей, Університетська книга, 2012. – 352 с.

ДОДАТОК А

Дані параметрів робочого приміщення, їх позначення та одиниці вимірювання

Таблиця 1.1 – Параметри робочого приміщення

№	Параметри	Позначення	Значення	Одиниці вимірювання
1	Довжина робочого приміщення	a	9,0	м
2	Ширина робочого приміщення	b	6,0	м
3	Висота робочого приміщення	H	3,2	м
4	Ширина вікна	c	1,4	м
5	Висота вікна	d	1,9	м
6	К-ть вікон	$n_в$	3	шт.
7	Висота верхнього краю вікна відносно умовної робочої поверхні	h	2,2	м
8	Висота карнизу протилежної будівлі відносно підвіконня	H'	12,5	м
9	Відстань до затіняючого будинку	D	23	м
10	Відстань розрахункової точки (робочої поверхні) до зовнішньої стіни	l	5,0	м

ДОДАТОК Б

План евакуації на підприємстві з позначеннями

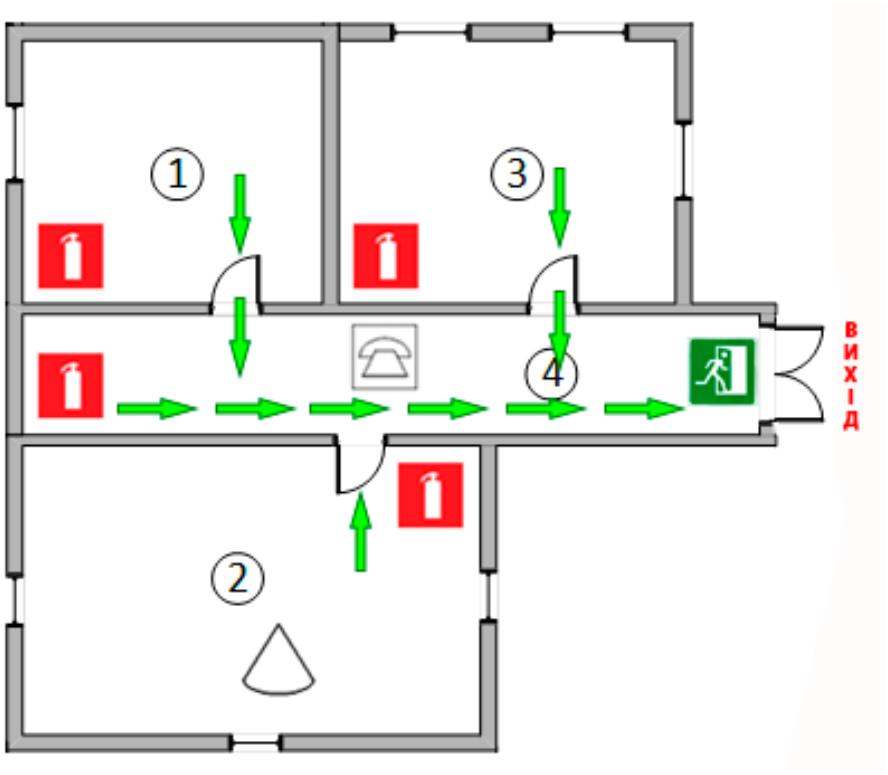


Рисунок 2.1 – План евакуації

Таблиця 2.1 – Позначення на плані евакуації

Позначення	Опис
	Місце розташування вогнегасників
	Місце розташування телефонного апарату
	Знак просторової орієнтації «Ви перебуваєте тут»
	Стрілка шляху евакуації
	Евакуаційний вихід

ЗВІТ

про перевірку на унікальність кваліфікаційної роботи бакалавра на тему:

«Геймифікований трекер звичок»

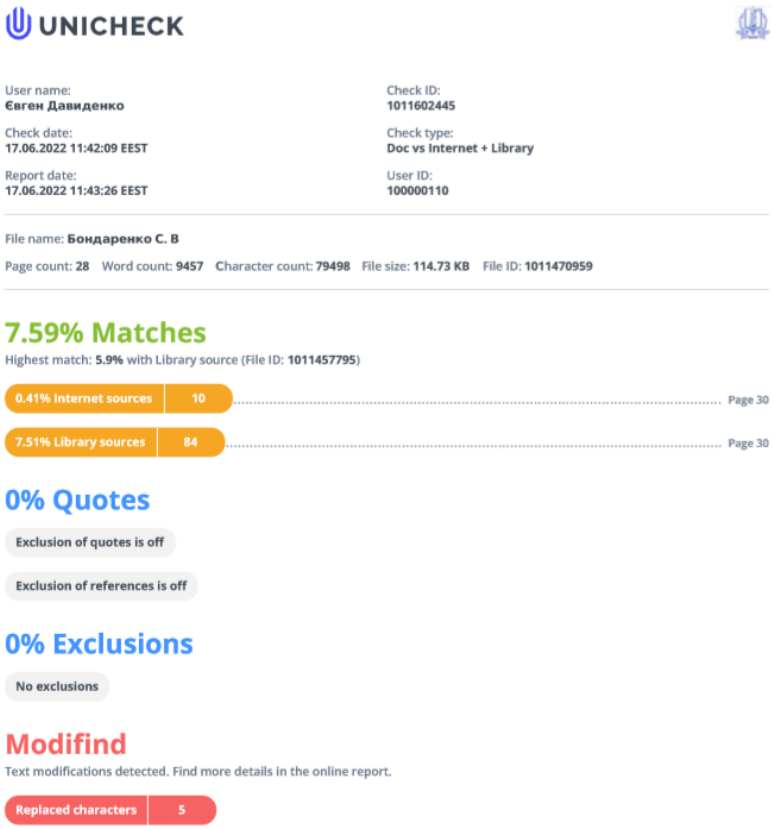
студентки спеціальності 121 «Інженерія програмного забезпечення», 409 групи

Бондаренко Стефанії Віталіївни

прізвище, ім’я, по батькові

Перевірку тексту здійснено сервісом: онлайн-сервіс Unicheck.

Результат перевірки тексту кваліфікаційної роботи бакалавра: схожість складає 7,59%.



UNICHECK

User name: Євген Давиденко
Check ID: 1011602445
Check date: 17.06.2022 11:42:09 EEST
Check type: Doc vs Internet + Library
Report date: 17.06.2022 11:43:26 EEST
User ID: 100000110

File name: Бондаренко С. В

Page count: 28 Word count: 9457 Character count: 79498 File size: 114.73 KB File ID: 1011470959

7.59% Matches

Highest match: 5.9% with Library source (File ID: 1011457795)

0.41% Internet sources 10 Page 30

7.51% Library sources 84 Page 30

0% Quotes

Exclusion of quotes is off

Exclusion of references is off

0% Exclusions

No exclusions

Modifind

Text modifications detected. Find more details in the online report.

Replaced characters 5

Студентка:

_____ С. В. Бондаренко

підпис ініціали, прізвище

Керівник:

завідувач кафедри ІПЗ, канд. техн. наук, доцент

_____ Є. О. Давиденко

підпис ініціали, прізвище

Дата: «___» _____ 2022 р.

АНОТАЦІЯ

до кваліфікаційної роботи бакалавра

«Геймифікований трекер звичок»

Студентка 409 гр.: Бондаренко Стефанія Віталіївна

Керівник: канд. техн. наук, доцент Давиденко Є. О.

Дана робота присвячена розробці програмного забезпечення для автоматизації процесу формування звичок з підвищенням власної ефективності та самомотивації, ґрунтуючись на роботі з базами даних: коучів, методичних рекомендацій та досягнень.

Об’єкт роботи: процес формулювання корисних звичок з підвищенням власної ефективності та самомотивації.

Предмет роботи: програмні засоби створення геймифікованого трекеру звичок.

Мета: автоматизація процесу формування звичок за рахунок розробки програмного забезпечення геймифікованого трекеру звичок.

Кваліфікаційна робота бакалавра складається з вступу, чотирьох розділів, висновків та переліку джерел посилання.

У вступі визначається актуальність теми, що приймається за мету та невеликий огляд поставленої задачі, предмет дослідження та об’єкт роботи.

У першому розділі описується аналітична частина, тобто огляд існуючих застосунків-аналогів трекерів звичок, визначення функціоналу, переваг та недоліків програмного забезпечення, обґрунтовується план виконання завдання. Наступною частиною розділу є формування та опис специфікації вимог до програмного забезпечення, що розробляється.

У другому розділі описується процес розробки проєктних рішень, що забезпечують виконання пунктів специфікації вимог до програмного забезпечення, тобто моделювання об’єкту та предмету дослідження, а також функціональні та інформаційні моделі програмного забезпечення. В результаті процесу розробки складено детальний алгоритм вирішення поставленої задачі.

У третьому розділі описується результат виконаної роботи з

конструювання та моделювання програмного забезпечення, включаючи вибір технологій, мови програмування, та компонентів застосунку, тобто бібліотек, плагінів та ін, розробку UML-діаграм та опис інтерфейсів геймифікованого трекеру звичок.

У четвертому розділі демонструється проведена робота з кодування та тестування застосунку геймифікованого трекеру звичок, крім того описується аналіз отриманих результатів під час тестування програмного забезпечення.

У висновках проводиться аналіз роботи та отриманих результатів.

Кваліфікаційна робота бакалавра викладена на 59 сторінок, вона містить 4 розділи, 36 ілюстрацій, 18 таблиць, 23 джерел в переліку посилань.

Ключові слова: *геймифікований трекер, формування звичок, створення вебзастосунку, розробка на Laravel, робота з коучем, підвищення ефективності та самомотивації, автоматизація процесу формування звичок, розробка програмного забезпечення.*

ABSTRACT

of the Bachelor's Thesis

"Gamified habits tracker"

Student: Bondarenko Stefaniia

Supervisor: Candidate of Technical Sciences (Ph. D.), Associate Professor

Davydenko Y. O.

This work is devoted to the development of software to automate the process of forming habits with increasing efficiency and self-motivation, based on working with databases: coaches, guidelines, and achievements.

The object of work: the process of formulating useful habits with increasing efficiency and self-motivation.

The subject of work: software tools for creating a gamified habit tracker.

Objective: to automate the process of habit formation through the development of gamified habit tracker software.

The qualification work of the bachelor consists of an introduction, four chapters, conclusions, and list of sources links.

The introduction determines the relevance of the topic to be taken as a goal and a brief overview of the task, the subject of research, and the object of work.

The first section describes the analytical part, ie an overview of existing applications-analogs of habit trackers, determining the functionality, advantages, and disadvantages of software, and substantiating the plan for the task. The next part of the section is the formation and description of the specification of the requirements for the software being developed.

The second section describes the process of developing design solutions that ensure compliance with the specification of software requirements, ie modeling of the object and subject of research, as well as functional and informational models of software. As a result of the development process, a detailed algorithm for solving the problem was developed.

The third section describes the results of the software design and modeling work, including the choice of technologies, programming languages, application components, ie libraries, plugins, etc., development of UML diagrams, and description of gamified habit tracker interfaces.

The fourth section demonstrates the work done on coding and testing the application of gamified habit tracker, in addition, describes the analysis of the results obtained during software testing.

The conclusions analyze the work and the results obtained.

The qualification work of the bachelor is presented on 59 pages, it contains 4 sections, 36 illustrations, 18 tables, 23 sources in the list of references.

Keywords: *gamified tracker, habit formation, web application creation, development on Laravel, work with a coach, increase of efficiency and self-motivation, automation of habit formation process, software development.*

ВІДГУК

на кваліфікаційну роботу бакалавра
студентки 409 групи ЧНУ ім. П. Могили
Бондаренко Стефанії Віталіївни
«Геймифікований трекер звичок»

Кваліфікаційна робота бакалавра здобувачки Бондаренко Стефанії Віталіївни присвячена автоматизації процесу формування звичок за рахунок розробки програмного забезпечення геймифікованого трекеру звичок.

Тема роботи гарно опрацьована, підібрані вдалі приклади аналогів, результат кваліфікаційної роботи повністю задовольняє поставленим завданням. Здобувачкою самостійно підібрана література та довідникові матеріали, що були уміло та професійно опрацьовані. Під час виконання кваліфікаційної роботи Бондаренко С. В. проявляла самостійність, гнучкість мислення та швидке вирішення проблем у критичних ситуаціях, самостійну ініціативу і активність у створенні нових функцій застосунку та роботі з проблемами, що виникали у процесі роботи.

Робота виконана якісно та на високому професійному рівні, а здобувачка Бондаренко Стефанія Віталіївна під час виконання роботи проявила здібності до практичного застосування набутих знань та навичок при реалізації поставлених завдань.

Кваліфікаційна робота «Геймифікований трекер звичок» може бути допущена до захисту, а її авторка Бондаренко Стефанія Віталіївна заслуговує присвоєння їй освітньої кваліфікації «Бакалавр з інженерії програмного забезпечення» за спеціальністю 121 «Інженерія програмного забезпечення» галузі знань 12 «Інформаційні технології».

Керівник кваліфікаційної роботи,
завідувач кафедри ІПЗ,
канд. техн. наук, доцент Є. О. Давиденко

РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студентки 409 групи ЧНУ ім. П. Могили

Бондаренко Стефанії Віталіївни

«Геймифікований трекер звичок»

Кваліфікаційна робота бакалавра студентки Бондаренко Стефанії Віталіївни присвячена розробці вебзастосунку геймифікованого трекеру звичок, метою якого є автоматизація процесу набуття звичок, задля чого програмне забезпечення має відповідний функціонал.

Робота виконана у відповідності до завдання, в повному обсязі у встановлений термін та складається з чотирьох тематичних розділів та спеціальної частини з охорони праці. У вступі визначається актуальність теми, що приймається за мету та невеликий огляд поставленої задачі, предмет дослідження та об’єкт роботи.

У першому розділі розглядається аналітична частина, тобто огляд існуючих застосунків-аналогів трекерів звичок, визначення функціоналу, переваг та недоліків програмного забезпечення, обґрунтовується план виконання завдання. Наступною частиною розділу є формування та опис специфікації вимог до програмного забезпечення, що розробляється.

У другому розділі моделюється процес розробки проєктних рішень, що забезпечують виконання пунктів специфікації вимог до програмного забезпечення, тобто моделювання об’єкту та предмету дослідження, а також функціональні та інформаційні моделі програмного забезпечення. В результаті процесу розробки описано детальний алгоритм вирішення поставленої задачі.

У третьому розділі демонструється результат виконаної роботи з конструювання та моделювання програмного забезпечення, включаючи вибір технологій, мови програмування, та компонентів застосунку, тобто бібліотек, плагінів та ін, розробку UML-діаграм та опис вебзастосунку.

У четвертому розділі демонструється проведена робота з кодування та тестування застосунку геймифікованого трекеру звичок, крім того описується

аналіз отриманих результатів під час тестування програмного забезпечення. У висновках проводиться аналіз роботи та отриманих результатів.

Враховуючи вищенаведене, вважаю, що кваліфікаційна робота «Геймифікований трекер звичок» виконана на професійному рівні і заслуговує високої оцінки, а її авторка Бондаренко Стефанія Віталіївна заслуговує присвоєння їй освітньої кваліфікації «Бакалавр з інженерії програмного забезпечення» за спеціальністю 121 «Інженерія програмного забезпечення» галузі знань 12 «Інформаційні технології».

Доцент кафедри ІІС,
канд. фіз.-мат. наук А. І. Воробйова

Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення
Ступінь вищої освіти "Бакалавр"
Спеціальність 121 "Інженерія програмного забезпечення"

Кваліфікаційна робота бакалавра

ГЕЙМИФІКОВАНИЙ ТРЕКЕР ЗВИЧОК

Виконала – студентка гр. 409 Бондаренко Стефанія Віталіївна
Керівник – канд. техн. наук, доцент Давиденко Євген Олександрович
Консультант – канд. техн. наук, доцент Алексєєва Анна Олександрівна

Миколаїв – 2022

Інженерія
Програмного
Забезпечення

Актуальність теми КРБ

Зумовлена сучасними тенденціями особистісного самовдосконалення. Для отримання результату важливо займатися справою регулярно та приділяти достатню кількість часу.

Методика набуття звички хоч і проста, але в сучасному житті є проблема розподілу часу та незручності ведення графіку справ. Саме для цього використовуються трекери звичок, вони допомагають керувати своїм часом з користю, оскільки в них

можна додавати звички та керувати ними, відмічати прогрес та відслідковуючи регулярність виконання поставлених цілей,

а головне – це зручно та мобільно, також додатково можна скористуватися допомогою коуча та отримати досягнення

в застосунку, що дуже допомагає та мотивує.

ОБ'ЄКТ РОБОТИ:

процес формування корисних звичок з підвищенням власної ефективності та самомотивації.

ПРЕДМЕТ РОБОТИ:

програмні засоби створення геймифікованого трекеру звичок.

МЕТА:

автоматизація процесу формування звичок за рахунок розробки програмного забезпечення геймифікованого трекеру звичок.

ЗАВДАННЯ:

- аналіз застосунків-аналогів;
- відокремлення проблем та пошук їх вирішення;
- визначення функціоналу застосунку;
- створення блок-схем алгоритму роботи геймифікованого трекеру звичок та UML-діаграм;
- розробка дизайну досягнень та макету застосунку;
- збір та аналіз інформації методичних рекомендацій;
- розробка front-end частини вебзастосунку на базі технологій Bootstrap, Vue.js;
- розробка back-end частини вебзастосунку на базі технологій Laravel, MySQL.

Аналіз застосунків-аналогів

Назва	Balance	Habitbull	Habitshare
Виробник	Приватна компанія *Elevate Labs*	Приватна компанія *App Holdings*	Приватна компанія *Luke Bickston*
Функції	1. Відповідати на питання щодо особистого досвіду, медітації, цілей, проблем. 2. Аудіотека з більш ніж 1000 файлів для медитації. 3. Застосунок персоналізується і підбирає більш ефективні рекомендації для кожного користувача. 4. Бібліотека матеріалів для медитації (збірки практик, порад, музики, ігри). 5. Інтерактивний сингл Wind Dow. Імерсийні медитації.	1. Спілкування на спеціалізованих форумах. 2. Можливість увімкнути нагадування щодо нових звичок. 3. Експорт своїх даних у форматі CSV. 4. Віжет для екрана (Лише на Android). 5. Створення будь-яких цілей. 6. Перевірка та вивчення особистих даних за допомогою інстументів глибокої аналітики. 7. Синхронізація декількох пристроїв.	1. Запланувати нагадування, щоб пам'ятати про нові звички. 2. Створення власних навичок, що відповідають особистим конкретним цілям, включаючи щоденні та щотижневі звички. 3. Додавання друзів для додаткової відповідальності та мотивації.
Недоліки	1. Застосунок спрямований лише на формування звичок, що стосуються медитативних практик, є платний функціонал.	1. Застарілий інтерфейс, інколи складно знайти необхідний функціонал, оскільки вони згруповані не завжди асоціативно зрозуміло, деякі функції приховано в меню.	1. Відсутність мотиваторів та теоретичної інформації для користувачів 2. Не можна синхронізувати декілька пристроїв.
Вебсайт	www-balanceapp-com	www-habitbull-com	https://habitshareapp.com

Специфікація Вимог до ПЗ

Ролі користувачів

- коуч;
- адміністратор;
- користувач.

Вимоги до технічного забезпечення

- процесор з підтримкою SSE2;
- оперативна пам'ять 8 ГБ;
- доступ до мережі Інтернет.

Вимоги до програмного забезпечення

- ОС Windows (10 та вище версії)
- ОС iOS та Android (вище версії 7.1)

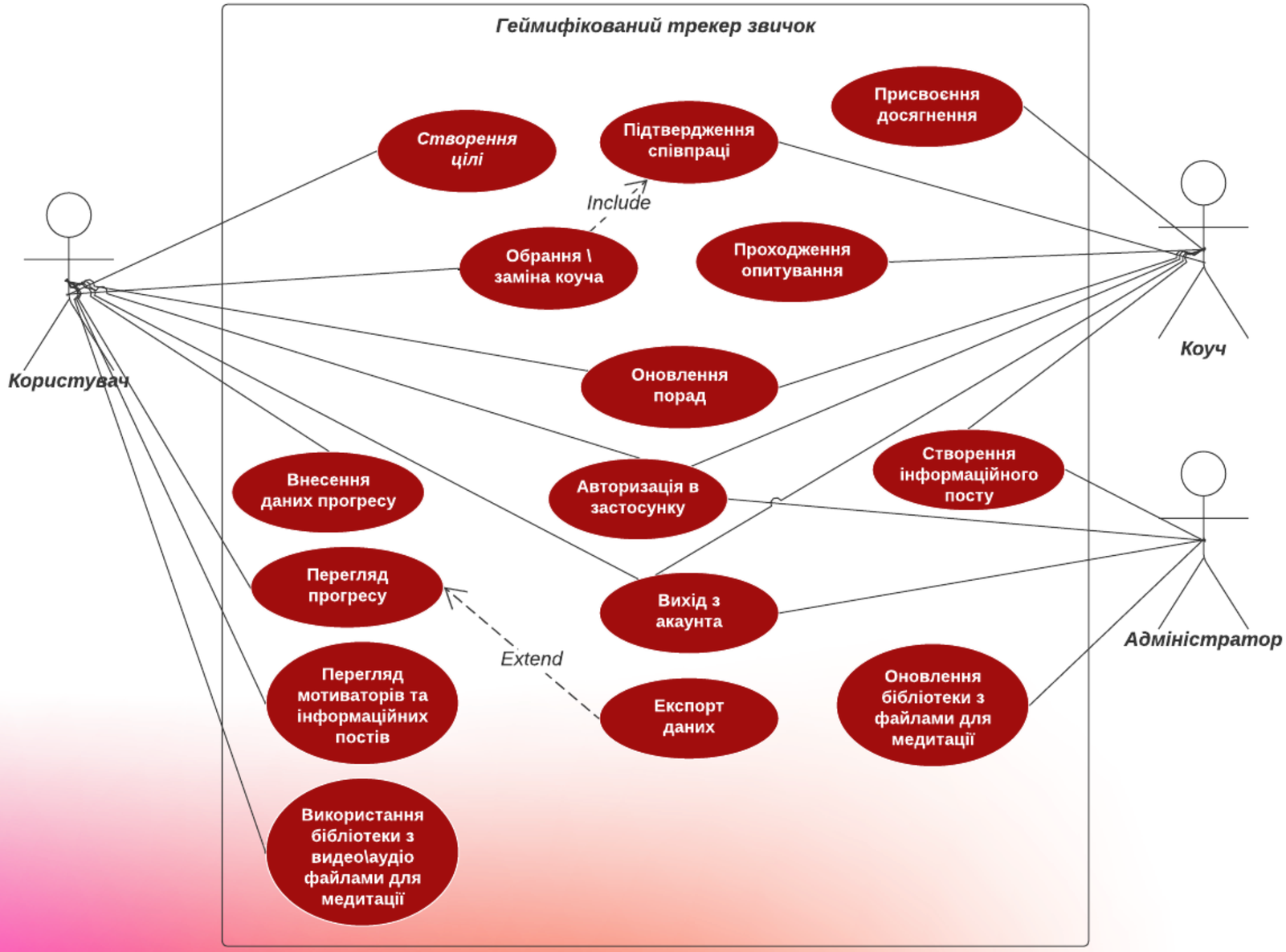
Призначення ПЗ

Автоматизація процесу формування звичок за рахунок розробки ПЗ геймифікованого трекеру звичок.

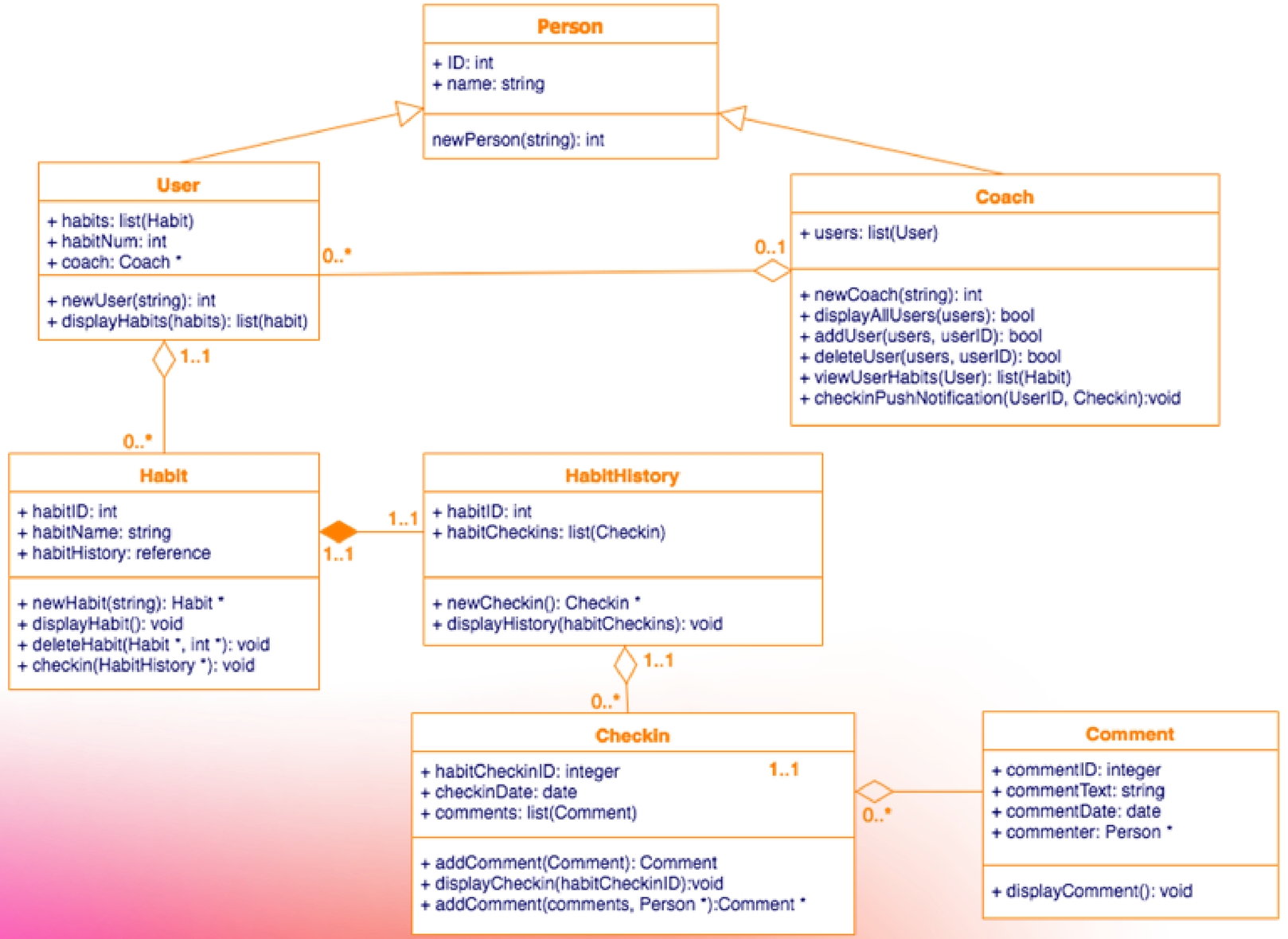
Основні функції

1. реєстрація користувачів;
2. авторизація користувачів;
3. розміщення та пошук мотиваторів;
4. додавання цілей;
5. вибір коуча;
6. відмічання прогресу виконання цілей;
7. допомога з медитативними практиками (бібліотека матеріалів, таймери і т.д.);
8. перегляд та даних прогресу;
9. чат з коучем;
10. історія звичок;
11. календар з актуальними та завершеними звичками.

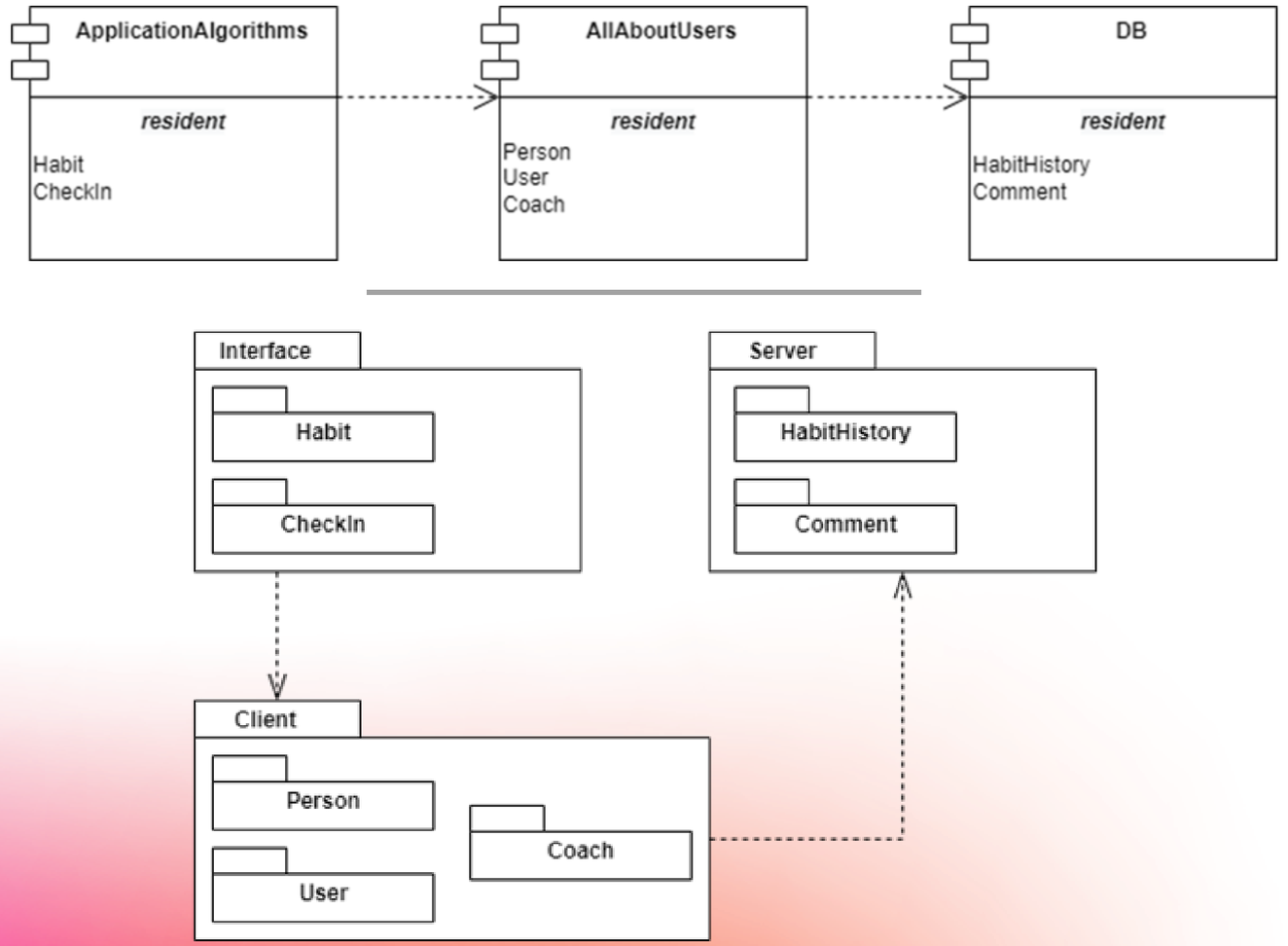
Use Case


Геймифікований трекер звичок
Створення цілі
Підтвердження співпраці
Присвоєння досягнення
Include
Обрання \ заміна коуча
Проходження опитування
Оновлення порад
Внесення даних прогресу
Авторизація в застосунку
Створення інформаційного посту
Перегляд прогресу
Вихід з акаунта
Extend
Перегляд мотиваторів та інформаційних постів
Експорт даних
Оновлення бібліотеки з файлами для медитації
Використання бібліотеки з видео\аудіо файлами для медитації
Користувач
Коуч
Адміністратор

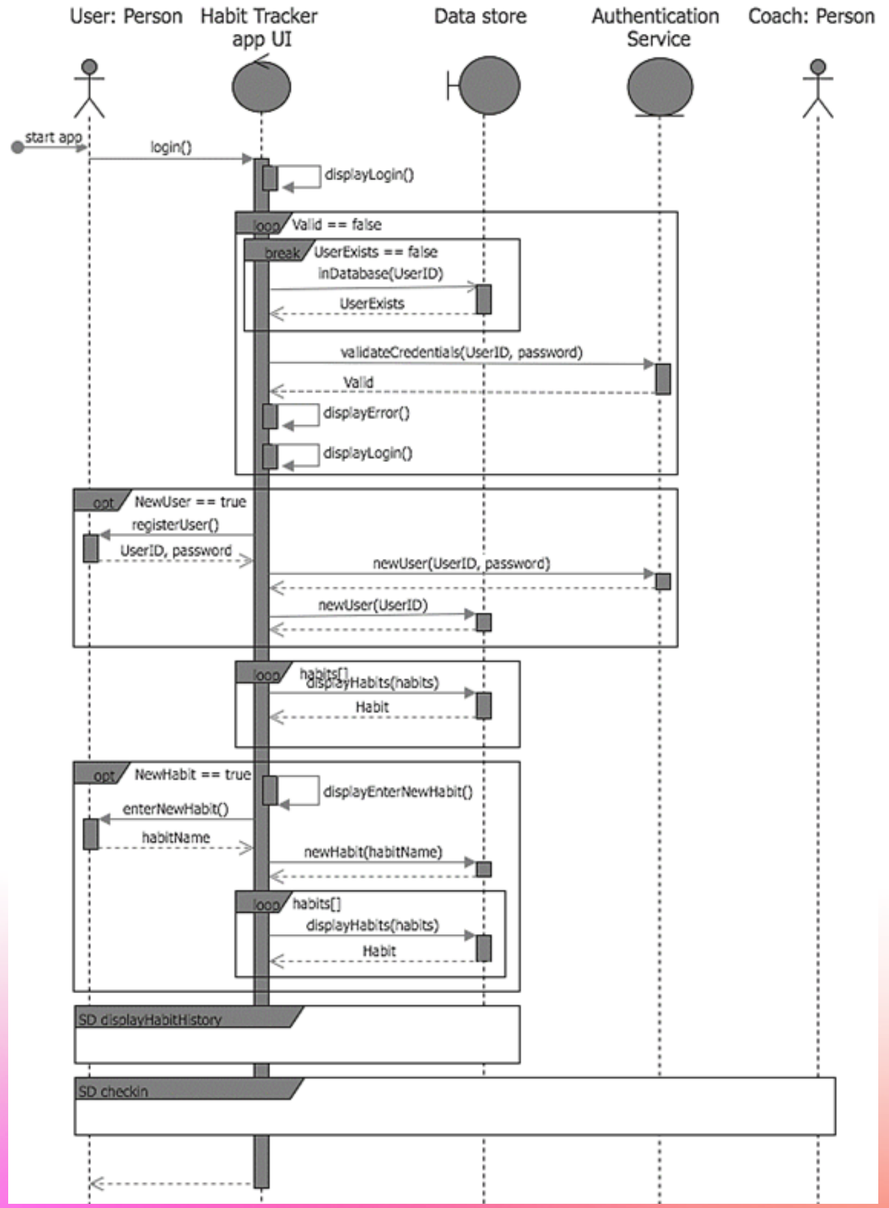
Діаграма класів

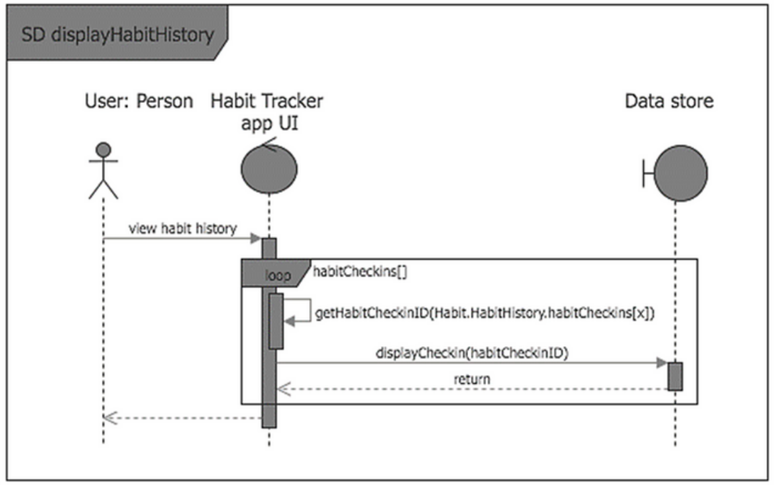


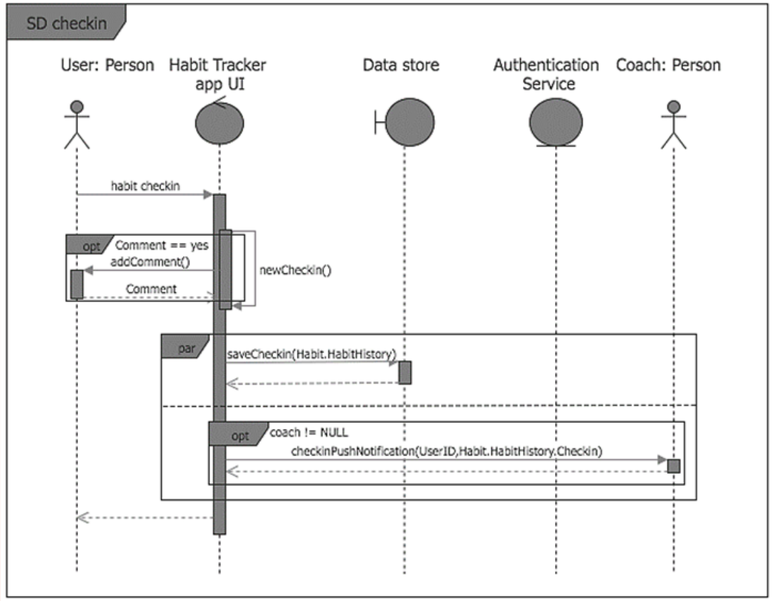
Діаграма компонентів та пакетів



Діаграми взаємодії







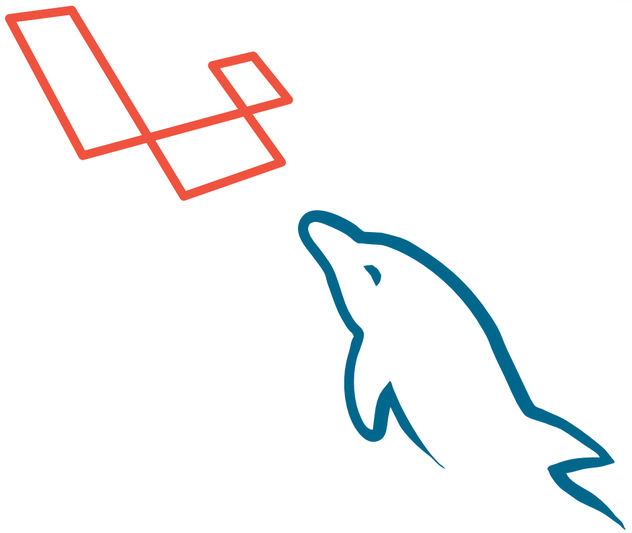
Стек технологій

МОВИ ПРОГРАМУВАННЯ

PHP та JavaScript

FRONT-END

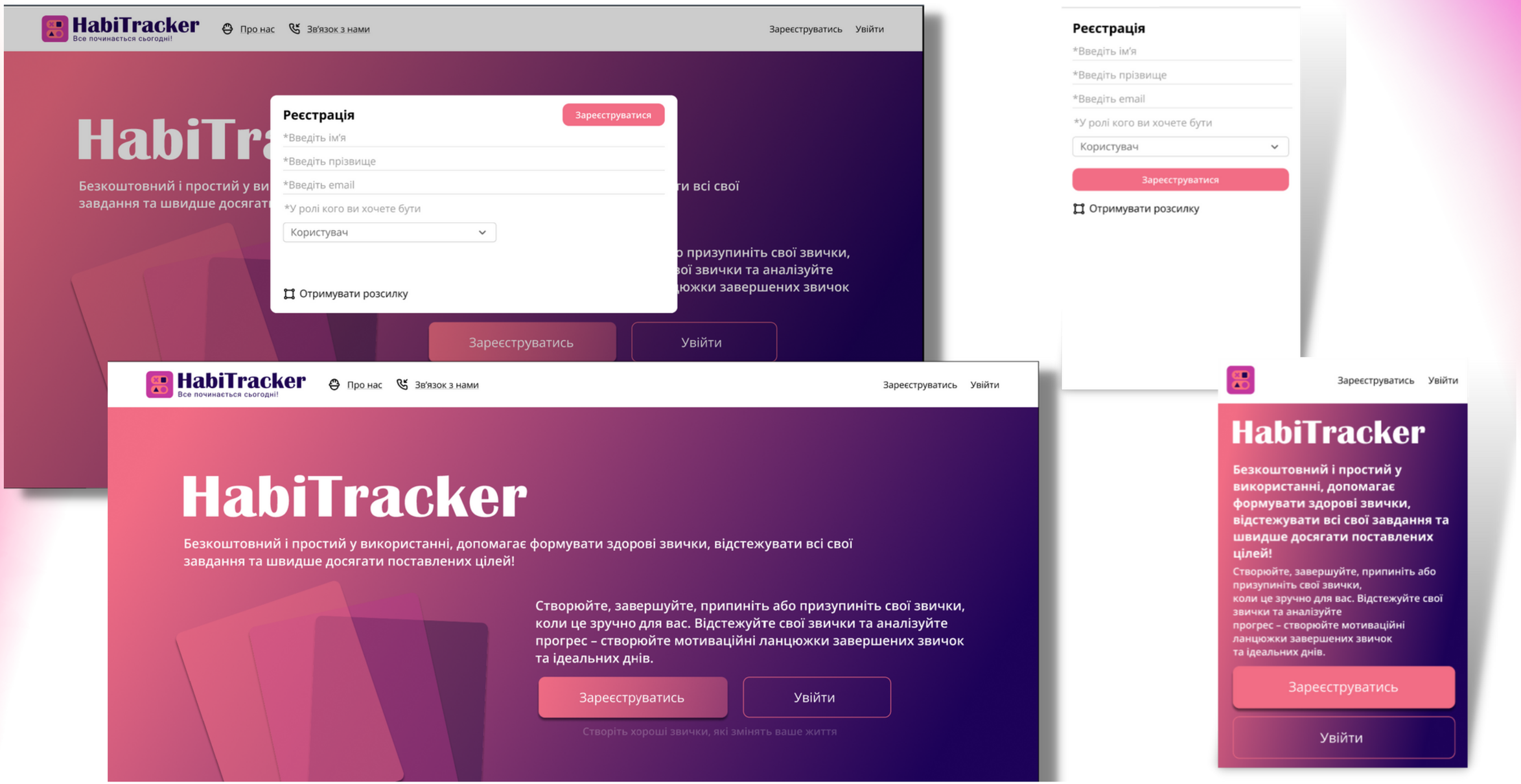
Bootstrap та Vue.js



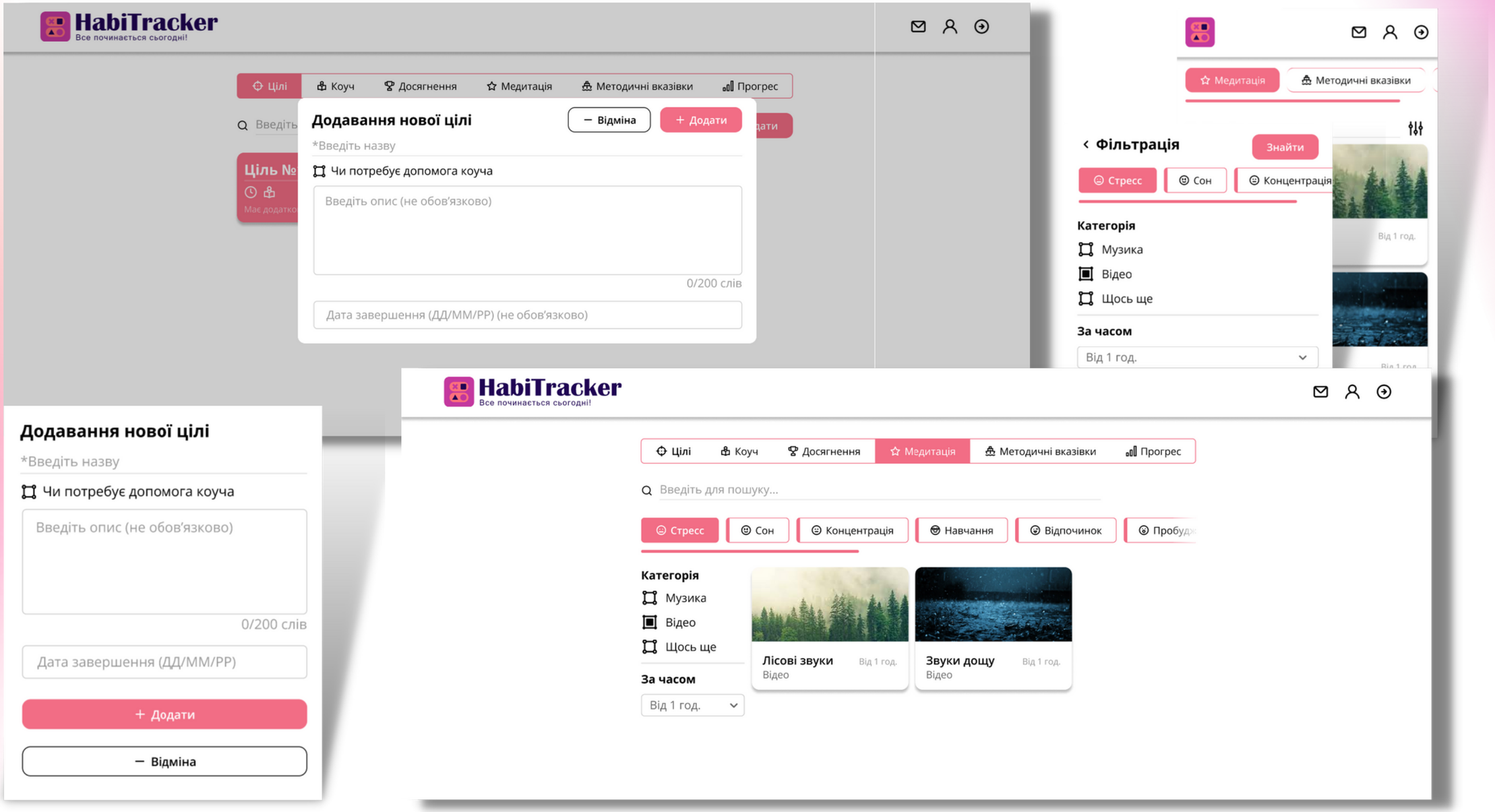
BACK-END

Laravel та MySQL

Дизайн застосунку



Дизайн застосунку



Відео з демонстрацією роботи застосунку

Висновки

У ході виконання кваліфікаційної роботи бакалавра автоматизовано процес формування звичок за рахунок розробки програмного забезпечення – вебзастосунку геймифікованого трекеру звичок.

Для досягнення визначеної мети вирішено поставлені завдання:

-проаналізовано предметну область та застосунки-аналоги;

-відокремлено проблеми та знайдено рішення їх усунення;

-визначено функціонал застосунку;

-створено блок-схеми алгоритму роботи геймифікованого трекеру звичок та необхідні UML-діаграми;

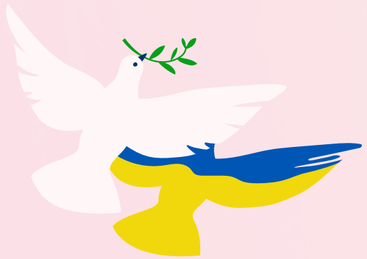
-розроблено дизайну досягнень та макету застосунку;

-зібрано та проаналізовано інформацію методичних рекомендацій;

-розроблено front-end частину вебзастосунку на базі технології Vue.js, Bootstrap;

-розроблено front-end частину вебзастосунку на базі технології Laravel, MySQL.

Результатом проведеної роботи є вебзастосунок геймифікований трекер звичок "HabiTracker".



ДЯКУЮ ЗА УВАГУ!

ГОТОВА ВІДПОВІСТИ НА ВАШІ ПИТАННЯ.