

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

(повна назва)

Кафедра Системотехніки

(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Проектування та реалізація мобільного застосунку для підтримки формування корисних звичок

(тема роботи)

Виконав:

здобувач 2 року навчання,

групи ІТПМ-24-1

Шалаєв Є. Р.

(прізвище, ініціали)

Спеціальність 122 – Комп'ютерні науки

(код і повна назва спеціальності)

Тип програми Освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційні технології проектування

(повна назва освітньої програми)

Керівник проф. каф. СТ. Перова І.Г.

(посада, прізвище, ініціали)

Допускається до захисту
Завідувач кафедри

(підпис)

Гребеннік І.В.

(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Системотехніки

Рівень вищої освіти другий (магістерський)

Спеціальність 122 – Комп'ютерні науки

Тип програми Освітньо-професійна

Освітня програма Інформаційні технології проектування

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Шаласву Єгору Романовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Проектування та реалізація мобільного застосунку для підтримки формування корисних звичок

затверджена наказом університету від 24 листопада 2025 р. № 1058

2. Термін подання здобувачем роботи до екзаменаційної комісії 11 грудня 2025 р.

3. Вихідні дані до роботи розробити компоненти інформаційної системи для сервісу відстеження звичок у вигляді мобільного додатку, використовуючи елементи гейміфікації, та реалізувати усі необхідні бізнес-функції. Розробити інтерфейс для системи, завдяки якому користувачі можуть мати доступ до всіх функцій діяльності. Перелік використаних програмних засобів: Microsoft Windows 11, IntelliJ IDEA, Visual Paradigm, MySQL Workbench Allfusion Process Modeler.

4. Перелік питань, що потрібно опрацювати в роботі 1 Аналіз предметної області дослідження, 1.1 Аналіз предметної області, 1.2 Огляд існуючих розробок та підходів, 1.3 Вибір програмної платформи та середовища розробки, 1.4 Визначення сфери застосування розроблювальної системи, 1.5 Призначення розробки, 1.6 Мета створення системи, 1.7 Основні функціональні можливості системи, 2 Дослідження існуючих підходів, методів, засобів, технологій, інструментарію для вирішення подібних задач, 2.1 Огляд наукових підходів до формування звичок, 2.2 Гейміфікація як інструмент мотивації, 2.3 Використання гейміфікації в мобільних системах підтримки здоров'я, 2.4 Технологічні підходи у реалізації систем формування звичок, 2.5 Висновки дослідження, 3 Постановка задачі дослідження, 4 Розробка методу (інформаційної технології) вирішення задачі, 4.1 Розробка системних вимог до системи, 4.2 Розробка функціональної моделі, 4.3 Розробка моделі потоків даних системи, 4.4 Визначення функціональних вимог до клієнтської сторони мобільного застосунку, 5 Опис прийнятих

проектних рішень при розробці системи, 5.1 Логічне і фізичне моделювання даних, 5.2 Створення бази даних на платформі СУБД, 5.3 Розробка алгоритму роботи системи, 5.4 Тестування розробленого програмного забезпечення, 5.5 Розробка тригерів серверної частини системи


5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій _____

КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів роботи | Строк/термін виконання етапів роботи | Примітка |
|----|--|--------------------------------------|----------|
| 1 | Аналіз систем з реалізованими прототипами функції | 24.11.25-25.11.25 | виконано |
| 2 | Дослідження існуючих підходів | 25.11.25-26.11.25 | виконано |
| 3 | Визначення сфери застосування розроблюваної системи | 26.11.25-27.11.25 | виконано |
| 4 | Постановка задачі | 27.11.25-28.11.25 | виконано |
| 5 | Розробка системних вимог до системи | 28.11.25-30.11.25 | виконано |
| 6 | Визначення функціональних вимог до клієнтської сторони додатку | 30.11.25-01.12.25 | виконано |
| 7 | Розробка моделі потоків даних системи | 01.12.25-03.12.25 | виконано |
| 8 | Створення бази даних на платформі СУБД | 03.12.25-04.12.25 | виконано |
| 9 | Розробка алгоритму роботи системи | 04.12.25-05.12.25 | виконано |
| 10 | Розробка тригерів | 05.12.25-06.12.25 | виконано |
| 11 | Тестування розробленого програмного забезпечення | 06.12.2025-10.12.2025 | виконано |

Дата видачі завдання 24 листопад 2025 р.

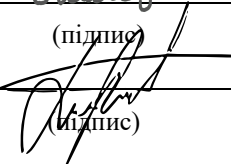
Здобувач _____


(підпис)

Шалаєв Є.Р.

(прізвище, ініціали)

Керівник роботи _____


(підпис)

Проф. каф. СТ Перова І.Г.

(посада, прізвище, ініціали)

РЕФЕРАТ

Звіт з практичної роботи: 125 сторінка, 48 рисунок, 35 джерел, 2 таблиці, 11 лістингів, 3 додатки.

БАЗА ДАНИХ, ГЕЙМІФІКАЦІЯ, ДІАГРАМИ, ДОДАТОК, ІНТЕРФЕЙС, ІНФОРМАЦІЙНА СИСТЕМА, МОТИВАЦІЯ, JAVA, MYSQL, SQL.

Об'єктом дослідження є процес формування та підтримки корисних звичок користувачів за допомогою цифрових інформаційних технологій.

Предметом дослідження є методи та інструменти проектування і реалізації мобільних інформаційних систем, орієнтованих на підтримку формування звичок, включно з компонентами гейміфікації.

Метою дослідження є проектування та розробка мобільного застосунку, що забезпечує створення й управління звичками, відстеження їх виконання, генерування мотиваційних сповіщень та інтерактивну гейміфіковану взаємодію з користувачем.

Методи досліджень – комплекс підходів та інструментів програмної інженерії, що включає методології функціонального та об'єктно-орієнтованого проектування, UML-моделювання, CASE-засоби для побудови структурних та поведінкових моделей, методи розробки реляційних баз даних, підходи до розробки мобільних застосунків у середовищі Android Studio, а також аналіз сучасних гейміфікаційних методик для підвищення мотивації користувачів.

У роботі здійснено аналіз існуючих мобільних рішень у сфері формування звичок, виявлено їхні ключові переваги та недоліки, а також запропоновано удосконалену модель, що включає інтеграцію гейміфікаційних механік для збільшення залученості та тривалості використання застосунку.

Сфера застосування розробленого рішення полягає в управлінні щоденними звичками та підвищенні особистої ефективності. Мобільний застосунок може використовуватися широкою аудиторією – від студентів і працівників офісів до людей, які прагнуть покращити свій розпорядок дня, дисципліну або психологічне благополуччя.

ABSTRACT

Work contains: 125 pages, 48 drawings, 35 links, 2 tables, 11 listings, 3 appendix.

APPLICATION, DATABASE, DIAGRAMS, GAMIFICATION, INTERFACE, INFORMATION SYSTEM, JAVA, MOTIVATION, MYSQL, SQL.

The object of the investigation is the process of forming and supporting the production of valuable signals using digital information technologies.

The subject of research is the methods and tools for designing and implementing mobile information systems that are oriented toward the support of clicker forming, including gamification components.

The method of further research is the design and development of a mobile application that will ensure the creation of sound management, their activation, generation of motivational messages and interactive gamified interaction with the correspondent.

Tracking methods are a set of approaches and software engineering tools, which includes methodologies of functional and object-oriented design, UML modeling, CASE methods for generating structural behavioral models, methods for developing relational databases, approaches to developing mobile applications in the middle of Android Studio, as well as analysis of current gamification techniques to increase the motivation of contributors.

The work has carried out an analysis of the main mobile solutions in the technology market, identified their key advantages and shortcomings, and also developed an improved model that includes integration gamification mechanics to increase gain and loss of fatigue.

The scope of the developed solution lies in the management of waste signals and increased special efficiency. Mobile app can be used by a wide audience - from students and office workers to people who want to improve their daily routine, discipline or psychological well-being.

ЗМІСТ

| | |
|---|----|
| ЗМІСТ | 6 |
| СКОРОЧЕННЯ, УМОВНІ ПОЗНАЧКИ І ТЕРМІНИ..... | 8 |
| ВСТУП..... | 9 |
| 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ..... | 10 |
| 1.1 Аналіз предметної області..... | 10 |
| 1.2 Огляд існуючих розробок та підходів | 11 |
| 1.3 Вибір програмної платформи та середовища розробки | 20 |
| 1.4 Визначення сфери застосування розроблювальної системи..... | 23 |
| 1.5 Призначення розробки..... | 24 |
| 1.6 Мета створення системи..... | 25 |
| 1.7 Основні функціональні можливості системи | 26 |
| 2 ДОСЛІДЖЕННЯ ІСНУЮЧИХ ПІДХОДІВ, МЕТОДІВ, ЗАСОБІВ, ТЕХНОЛОГІЙ, ІНСТРУМЕНТАРІЮ ДЛЯ ВИРІШЕННЯ ПОДІБНИХ ЗАДАЧ | 29 |
| 2.1 Огляд наукових підходів до формування звичок | 29 |
| 2.2 Гейміфікація як інструмент мотивації..... | 29 |
| 2.3 Використання гейміфікації в мобільних системах підтримки здоров'я..... | 30 |
| 2.4 Технологічні підходи у реалізації систем формування звичок | 30 |
| 2.5 Висновки дослідження | 30 |
| 3 ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ..... | 32 |
| 4 РОЗРОБКА МЕТОДУ (ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ) ВИРІШЕННЯ ЗАДАЧІ | 34 |
| 4.1 Розробка системних вимог до системи | 34 |
| 4.2 Розробка функціональної моделі | 36 |
| 4.3 Розробка моделі потоків даних системи | 48 |
| 4.4 Визначення функціональних вимог до клієнтської сторони мобільного застосування | 49 |
| 5 ОПИС ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ ПРИ РОЗРОБЦІ СИСТЕМИ.. | 54 |
| 5.1 Логічне і фізичне моделювання даних | 54 |
| 5.2 Створення бази даних на платформі СУБД..... | 60 |
| 5.3 Розробка алгоритму роботи системи..... | 63 |
| 5.4 Тестування розробленого програмного забезпечення..... | 64 |
| 5.5 Розробка тригерів серверної частини системи..... | 68 |

| | |
|----------------------|----|
| ВИСНОВКИ..... | 72 |
| ПЕРЕЛІК ДЖЕРЕЛ | 73 |

СКОРОЧЕННЯ, УМОВНІ ПОЗНАЧКИ І ТЕРМІНИ

ІС – інформаційна система;

БД – база даних;

СУБД – система управління базами даних;

CASE – Computer-Aided Software Engineering;

UML – Unified Modeling Language;

IDEF0 – методологія функціонального моделювання;

IDEF1X – методологія моделювання даних (логічної та фізичної моделей);

SQL – Structured Query Language.

ВСТУП

Основною метою цього проекту є розробка мобільного застосунку для підтримки формування корисних звичок користувачів. Такий інструмент дозволить вирішити актуальні проблеми, з якими стикаються люди, що прагнуть покращити власну продуктивність, підтримувати мотивацію та формувати сталі позитивні звички. Розроблений застосунок забезпечить зручну та інтуїтивну навігацію та інтегровані засоби гейміфікації, що сприятимуть підвищенню залученості та довготривалій взаємодії користувачів із системою.

Ключовою особливістю розроблюваного рішення стане використання ігрових механік — віртуальних рослин, рівнів та досягнень. Завдяки такому підходу рутинні процеси перетворюються на цікавий і вмотивований досвід. Застосунок надаватиме користувачам можливість створювати та керувати власними звичками, отримувати нагадування та переглядати статистику, що дозволяє краще розуміти власну динаміку розвитку. Система також включатиме реєстрацію та авторизацію користувачів, що дозволить забезпечити персоналізований доступ до даних та синхронізацію прогресу на різних пристроях. Важливою складовою є модуль автоматичного надсилання сповіщень, який допоможе підтримувати регулярність виконання задач.

Актуальність теми зумовлена зростанням попиту на цифрові інструменти саморозвитку, особливо у контексті підвищеної уваги до психологічного благополуччя, продуктивності та менеджменту часу. У сучасних умовах мобільні додатки стають одним із ключових засобів підтримки здорових навичок, а гейміфікація довела свою ефективність у сфері мотивації та утримання користувачів.

Результатом цього проекту стане повноцінний мобільний застосунок, що забезпечує створення та управління звичками, відстеження прогресу, надсилання нагадувань, а також надає інтерактивну систему гейміфікації для підвищення мотивації користувача. Це дозволить створити ефективний інструмент саморозвитку, що відповідає сучасним вимогам та тенденціям ринку мобільних застосунків.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ

1.1 Аналіз предметної області

Сучасний етап розвитку інформаційних технологій характеризується активним упровадженням інформаційних систем (ІС) у всі сфери людської діяльності. Інформаційна система — це комплекс програмних, технічних, організаційних та інформаційних засобів, призначених для збирання, зберігання, обробки, аналізу та передачі даних з метою підтримки прийняття рішень. Її основними компонентами є база даних (БД), серверна частина (бекенд), клієнтська частина (фронтенд), а також користувацький інтерфейс, через який здійснюється взаємодія з користувачем.

Для забезпечення надійності та ефективності роботи сучасні ІС часто реалізуються у вигляді розподілених систем, де обчислення, зберігання та обробка даних відбуваються на різних вузлах мережі. Такий підхід підвищує продуктивність, дозволяє масштабувати систему під зростаючу кількість користувачів і забезпечує балансування навантаження між серверами. У веб- або мобільних додатках серверна частина відповідає за логіку роботи системи, доступ до бази даних, а клієнтська — за інтерфейс користувача та взаємодію з нею.

Інформаційні системи можна класифікувати за різними критеріями:

- за сферою застосування (освітні, медичні, фінансові, торговельні, соціальні тощо);
- за призначенням (управлінські, облікові, аналітичні, рекомендаційні, комунікаційні);
- за архітектурою (централізовані, розподілені, хмарні).

У межах цієї класифікації системи, що орієнтовані на електронну комерцію, освітні процеси або особистісний розвиток, займають окрему нішу. Для таких систем важливу роль відіграють рекомендаційні модулі, аналітика поведінки користувача, а також механізми утримання користувачів — гейміфікація, персоналізація, мотиваційні елементи.

У контексті особистісного розвитку актуальним напрямом є інформаційні системи для формування та підтримки корисних звичок. Такі мобільні застосунки допомагають користувачам ставити цілі, відслідковувати їх виконання, аналізувати прогрес та підтримувати мотивацію. Ці системи часто

працюють із особистими даними, тому повинні забезпечувати збереження приватності та синхронізацію даних між пристроями.

Одним із найважливіших аспектів у подібних системах є психологічна складова взаємодії з користувачем. Саме тут велику роль відіграє гейміфікація — застосування ігрових механік у неігрових контекстах. Гейміфікація базується на використанні таких елементів, як бали, рівні, нагороди, візуалізація прогресу, таблиці лідерів, колекції досягнень тощо. Її метою є створення відчуття задоволення, залучення та прогресу у процесі досягнення цілей.

Застосування гейміфікації у мобільних додатках для формування звичок довело свою ефективність. За результатами досліджень, такі елементи допомагають підвищити утримання користувачів у 2–3 рази у порівнянні з традиційними трекерами звичок, які не мають ігрової складової. Використання ігрових елементів формує у користувача поведінкову мотивацію — бажання продовжувати взаємодію із системою заради відчуття досягнення та винагороди.

Мобільний застосунок, розробка якого розглядається в цій роботі, належить до класу інформаційних систем особистісного розвитку, що реалізовані у вигляді розподіленої клієнт-серверної архітектури. Серверна частина розроблена у середовищі IntelliJ IDEA мовою Java, з використанням СУБД MySQL для зберігання даних користувачів, звичок та прогресу. Клієнтська частина реалізована у Android Studio, що забезпечує інтуїтивно зрозумілий мобільний інтерфейс та зручну взаємодію з користувачем. Основна концепція додатку — поєднання системи відслідковування звичок із візуальною метафорою “росту рослини”, що реалізує принцип гейміфікації у процесі саморозвитку.

1.2 Огляд існуючих розробок та підходів

На сьогодні на ринку існує низка мобільних застосунків, які допомагають користувачам формувати та підтримувати звички. До найпопулярніших належать Habitica, Fabulous, Loop Habit Tracker, Forest та інші. Кожен з них реалізує власну модель мотивації користувача та має свої особливості.

Таблиця 1.1 – Порівняння існуючих підходів у застосунках

| Застосунок | Основна концепція | Елементи гейміфікації | Сильні сторони | Недоліки |
|-----------------------------------|---|-------------------------------------|---|---|
| Habitica | Трекер звичок у формі RPG-гри | Рівні, досвід, нагороди, квести | Сильна гейміфікація, соціальні елементи | Надлишкова складність, неінтуїтивний UX |
| Fabulous | Формування ранкових та вечірніх рутин | Досягнення, нагадування | Психологічний підхід, приємний інтерфейс | Менше гнучкості для кастомних звичок |
| Loop Habit Tracker | Мінімалістичний трекер звичок | Статистика, графіки | Простота, відкритий код | Відсутність гейміфікації |
| Forest | Зосередження на завданні через вирощування дерева | Візуальна нагорода (дерево росте) | Ефективна візуалізація, проста взаємодія | Не універсальний — орієнтований лише на фокусування |
| Habitflow (мій застосунок) | Формування звичок через метафору “рослини” | Рівні, візуальний прогрес, нагороди | Поєднання гейміфікації та відслідковування звичок | Потребує подальшої оптимізації мотиваційних сценаріїв |

Для прикладу проаналізуємо скріншоти деяких застосунків. Найочевиднішою є проблема Loop Habit Tracker. Візуал застосунку зображено на рисунку 1.1.

8:16

Habits

| | THU 29 | WED 28 | TUE 27 | MON 26 | SUN 25 |
|---------------------|--------------|--------------|--------------|--------------|--------------|
| Wake up early | ✓ | ✓ | × | ✓ | × |
| Cook healthy dinner | ✓ | ✓ | × | × | × |
| Write journal | ✓ | × | ✓ | ✓ | × |
| Track time | ✓ | ✓ | ✓ | ✓ | ✓ |
| Meditate | ✓ | ✓ | ✓ | × | ✓ |
| Run | 0.9 miles | 1.2 miles | 1.3 miles | 0 miles | 0.5 miles |
| Read books | 50 pages | 38 pages | 65 pages | 100 pages | 10 pages |
| Learn French | ✓ | ✓ | × | ✓ | ✓ |
| Play chess | ✓ | ✓ | × | × | × |
| Practice guitar | × | × | × | ✓ | ✓ |
| Call a friend | × | ✓ | × | × | ✓ |

Рисунок 1.1 – Интерфейс Loop Habit Tracker

Хоч мінімалізм і є сильною стороною додатку, однак основною метою додатків для відслідковування звичок все ж є не просто відмітка виконаної звички. Розробники цього застосунку не намагалися мотивувати своїх користувачів, через що залученість у Loop Habit Tracker, скоріш за все, буде абсолютно низькою.

Іншим застосунком, який я аналізував, є Forest app. У ньому розробники підійшли набагато серйозніше до питання мотивації користувачів. Розглянемо інтерфейс застосунку на рисунку 1.2 – 1.3.

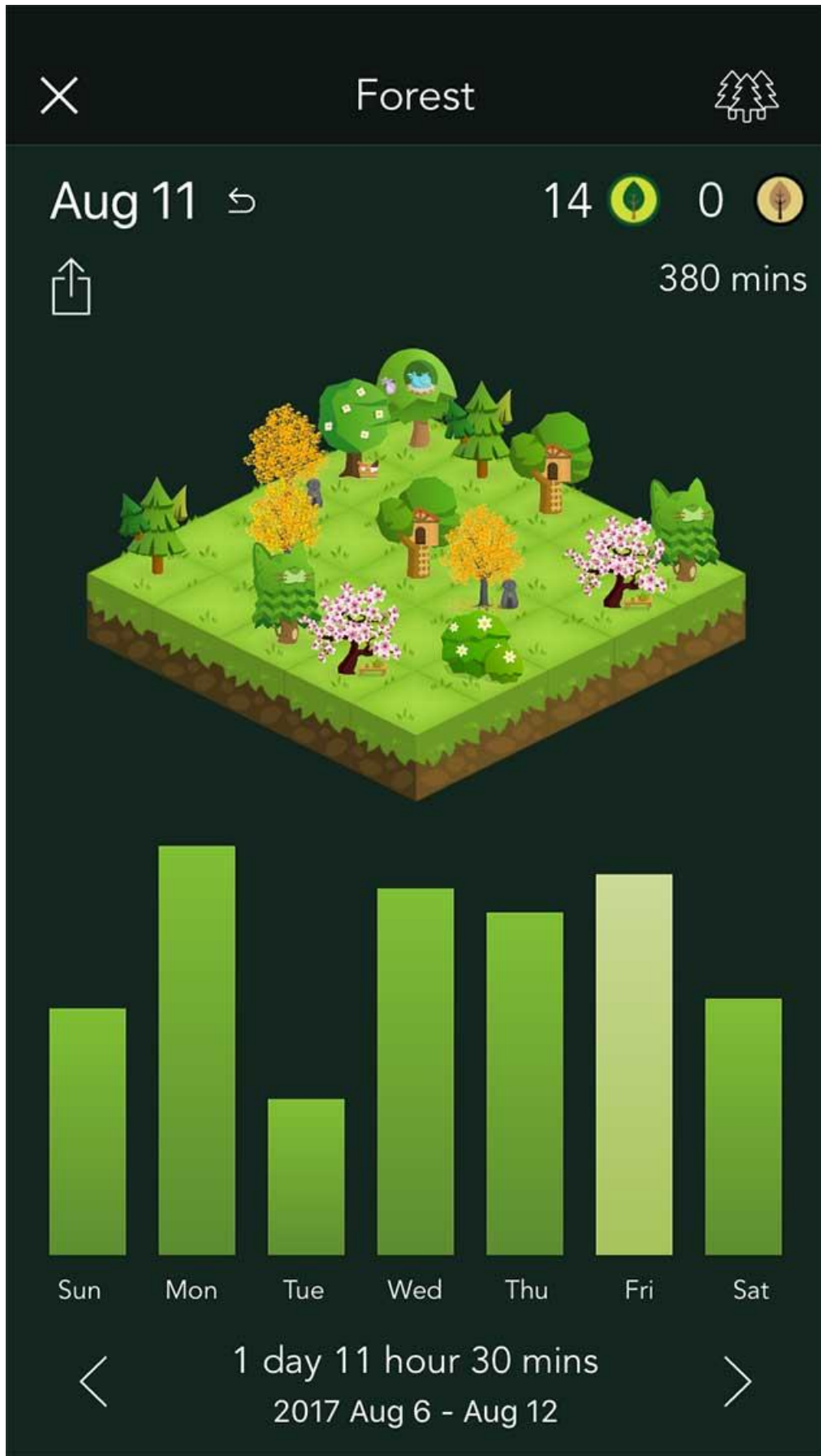


Рисунок 1.2 – Интерфейс Forest app

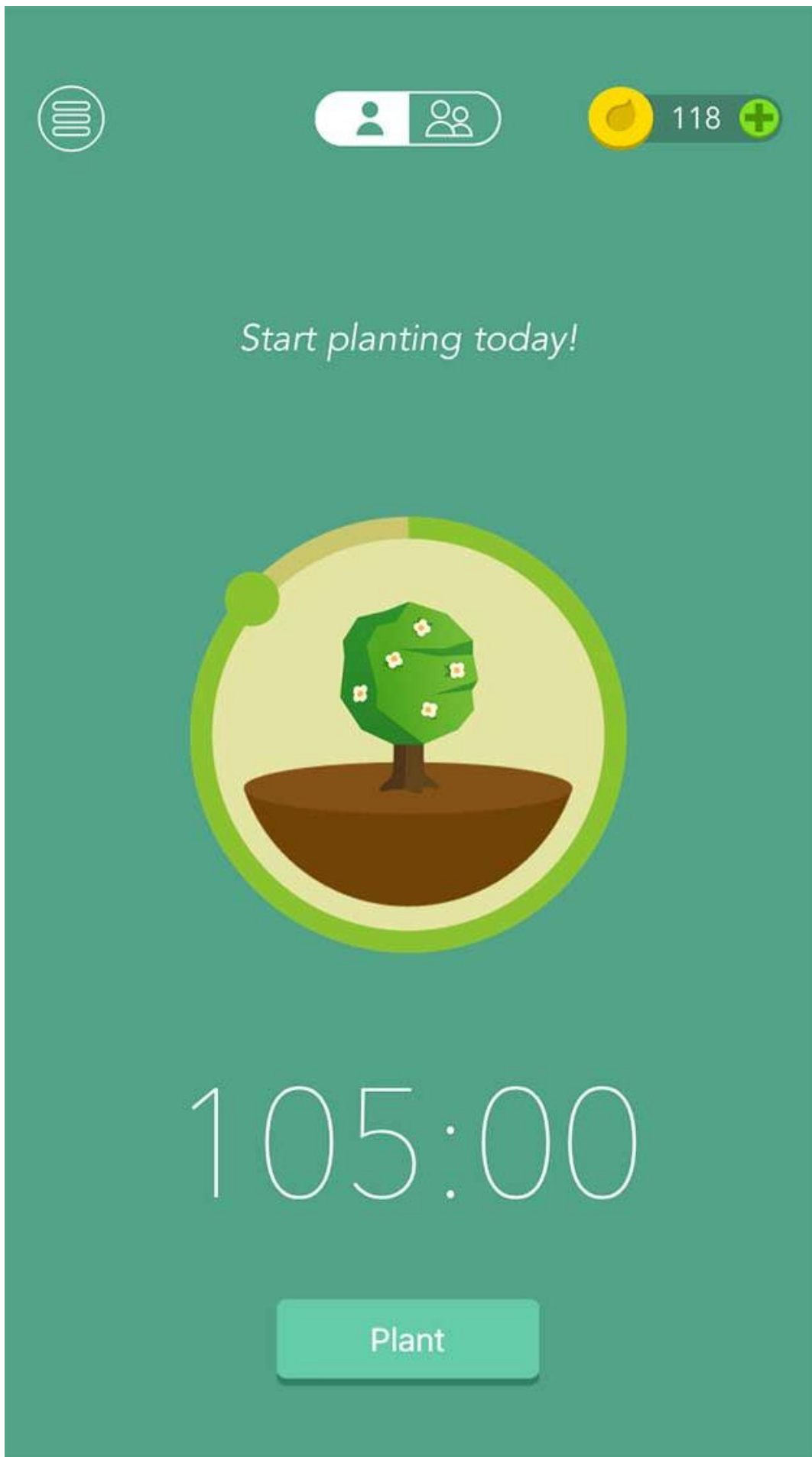



Рисунок 1.3 – Интерфейс Forest app


Перше, що легко помітити – пропрацьований візуал. Може здатися, що це – незначна деталь, однак окрім естетичного задоволення професійна робота з інтерфейсом допомагає зробити зрозумілий застосунок, який не буде відволікати користувачів та відлякувати їх складністю.


Сама ідея застосування гейміфікації доволі проста: Forest app – це, простими словами, таймер. Користувач виділяє обраний час(наприклад, пів години на прочитання книги) і запускає таймер. Якщо за цей час він відволікається(робить дзвінок, заходить до соціальних мереж, інших додатків), дерево, яке прив'язане до таймеру «помирає». Суть в тому, щоб мотивувати користувача «врятувати» дерево, а в майбутньому – побудувати ліс. Однак, як і було зазначено у таблиці, застосунок є дуже обмеженим. Він зорієнтований тільки на фокусування, тобто не має достатньої універсальності для відслідковування інших типів корисних звичок(кількісних, які потребують лічильника замість таймеру) чи позбавлення поганих. До того ж, ліс – єдиний елемент гейміфікації, що може дуже скоро набриднути користувачу.


Під кінець розглянемо Habitica – ще один додаток для відслідковування звичок. Цей застосунок є найближчим до розроблюваного мною HabitFlow, адже на відміну від інших додатків, його автори використовували величезну кількість елементів гейміфікації, фактично перетворивши нудний застосунок на гру, що, в теорії, дуже добре має мотивувати користувачів лишатися в додатку та працювати над своїми звичками. Вони використовують рівні, досвід, нагороди та квести, роблячи з Habitica гру у жанрі RPG(рольова гра). Однак, якщо подивитися на їхній інтерфейс на рисунку 1.4 – 1.5, одразу стає очевидною головна проблема.


Tasks Inventory Shops Party Group



Lydda Sneakthief 
 @FakeLemoness3 · Level 52 Rogue

 24 / 50







 615 / 1340

 134 / 134

Habits

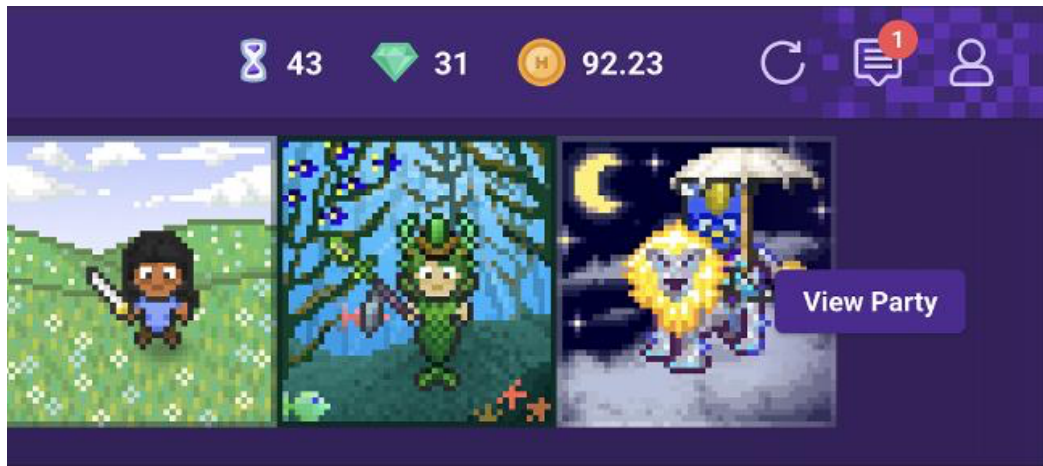
All Weak Strong

Add a Habit

-  15 minutes of cleaning  ▶▶ +7 | -1
-  1h Productive Work
Mark this Habit as done for each hour completed  ▶▶ 0
-  Eat Junk Food
C'mon, we need more fiber  ▶▶ 0

Daily

Рисунок 1.4 – Интерфейс Habitica



+ Add Task

Complete **Rewards** All Custom Wishlist

Add a Reward

- Watch one episode of a show
Rest and relax to some good content 20
- Treat yourself
Time to enjoy a little treat for a job well done! 10










| | | |
|---|--|---|
|  80 |  90 |  60 |
|  80 |  100 |  25 |
|  0 |  0 |  0 |

Рисунок 1.5 – Интерфейс Habitica

Подивившись на те, як виглядає застосунок, можна легко зрозуміти, що він розроблявся на дуже вузьку аудиторію. Особисто для мене, людини, що добре знайома з RPG-іграми, така ідея здається неймовірно цікавою. Такий застосунок я і сам міг би із задоволенням використовувати. Проблема в тому, що він використовує багато ігрових понять, різних валют, ресурсів та інших елементів. Вони дуже добре працюють на конкретну аудиторію молодих людей, особливо якщо ті вже мали досвід у подібних іграх. Однак якщо людина старшого віку, яка ніколи не грала у рольові ігри, або просто молода дівчина, якій таке не цікаво, спробують встановити цей застосунок, є дуже велика ймовірність, що дуже скоро їм набридне розбиратися в деталях гр-системи і вони захочуть обрати більш простий варіант. Тож основна проблема цього додатку – фокус лише на одній вузькій аудиторії.

Більшість аналізованих систем базується або на поведінковій психології, або на ігрових принципах мотивації. Однак існує проблема швидкого зниження залученості користувачів — після перших тижнів використання активність падає на 60–80%. Це свідчить про необхідність пошуку нових методів довготривалої мотивації.

У межах даної роботи пропонується поєднати два підходи:

1. Гейміфікацію — для емоційної мотивації через досягнення, прогрес та винагороду;
2. Візуалізацію росту — як інструмент позитивного підкріплення (кожна звичка «вирощує» власну рослину).

Такий підхід базується на поведінковій моделі формування звички (Habit Loop), що включає три елементи: сигнал (нагадування), дія (виконання звички) і винагорода (зростання рослини). Таким чином, користувач отримує не лише статистичний, а й емоційно-візуальний стимул, що формує стабільну поведінкову мотивацію.

Отже, проведений огляд існуючих розробок показав, що поєднання гейміфікації, поведінкової психології та візуалізації прогресу є найбільш перспективним напрямом для підвищення ефективності мобільних застосунків із підтримки формування корисних звичок. Це і стало основою концепції розроблюваного у цій роботі застосунку.

1.3 Вибір програмної платформи та середовища розробки

Для створення мобільного застосунку HabitFlow, який поєднує функціональність трекінгу звичок, елементи гейміфікації та персоналізовану взаємодію з користувачем, була обрана мова програмування Java та середовище розробки Android Studio. Такий вибір зумовлений поєднанням технічних переваг, зрілості екосистеми та відповідності вимогам до сучасних мобільних застосунків. Нижче подано розгорнуте пояснення причин використання Java саме для мобільної розробки в Android-середовищі.

Кросплатформеність у межах Android-екосистеми

Java багато років є основною мовою програмування для створення Android-застосунків. Усі офіційні SDK, бібліотеки, компоненти та системні інструменти Android були історично побудовані саме з урахуванням можливостей Java, що забезпечує високу сумісність та стабільність. Завдяки принципу "write once, run anywhere", Java-код у межах Android може працювати на будь-якому пристрої під керуванням Android — від бюджетних смартфонів до сучасних флагманів із різними процесорами та конфігураціями.

Для HabitFlow це означає можливість стабільної роботи на широкому спектрі пристроїв, незалежно від їх технічних характеристик чи версії ОС (за умови підтримки API-рівня).

Розвинена екосистема Android-бібліотек

Java має величезну екосистему інструментів та бібліотек, призначених саме для мобільної розробки. У контексті HabitFlow ключовими є:

- Android Jetpack Components (LiveData, ViewModel, Room, Navigation) — вони спрощують створення архітектури, керування станом, навігацією та локальною базою даних.
- Room Persistence Library — ORM-рішення для зручної роботи з локальною базою даних SQLite. HabitFlow використовує Room для збереження звичок, статистики виконання, прогресу користувача та налаштувань.
- AlarmManager / WorkManager — системні інструменти для створення нагадувань та фонових виконань задач, що є критично важливою частиною логіки трекера звичок.
- Firebase Services (за потреби) — для аутентифікації, хмарного збереження даних, push-сповіщень чи синхронізації.

Завдяки цим інструментам розробка значно спрощується, оскільки можна зосередитися на бізнес-логіці HabitFlow, а не на низькорівневій реалізації.

Висока продуктивність та стабільність роботи

Java запускається на Android Runtime (ART), який забезпечує оптимізоване виконання мобільного коду. ART попередньо компілює код у машинні інструкції, що зменшує затримки, робить роботу додатку плавною та знижує навантаження на процесор.

Для HabitFlow це особливо важливо, адже застосунок:

- багаторазово опрацьовує операції зі звичками протягом дня;
- виконує внутрішні обчислення, пов'язані з гейміфікацією (досвід, рівні, стрики);
- працює з локальною базою даних;
- може мати десятки активних нагадувань.

Завдяки оптимізації ART і особливостям Java, додаток працює стабільно, без підвисань і з мінімальним використанням ресурсів.

Безпека мобільних даних

Оскільки HabitFlow обробляє персональні дані користувача — звички, прогрес, щоденну активність, налаштування — безпека є одним із критичних аспектів.

Java забезпечує високий рівень безпеки у мобільних застосунках завдяки:

- автоматичному управлінню пам'яттю (захист від витоків пам'яті та переповнення буфера);
- інкапсуляції та суворій структурі коду;
- можливості шифрування локальних даних (AES, SHA тощо);
- можливості використання безпечного зберігання (EncryptedSharedPreferences, EncryptedFile);
- підтримці HTTPS, SSL, TLS.

Це дозволяє гарантувати, що дані HabitFlow будуть належним чином захищені навіть на стороні локального пристрою.

Масштабованість та адаптованість застосунку

Попри те, що HabitFlow є мобільним додатком, він має потенціал до зростання та розширення функціональності. Java та Android Studio дозволяють масштабувати мобільні застосунки як у частині бізнес-логіки, так і у частині роботи з даними.

У контексті HabitFlow це означає:

- можливість додавання нових функцій (чат-бот нагадувань, рекомендаційна система, рейтинги);
- розширення системи гейміфікації (квести, колекції, бонуси);
- підключення хмарних сервісів для синхронізації між кількома пристроями;

- адаптацію під планшети чи великі екрани.

Мобільні застосунки на Java можуть зростати без потреби в перенесенні або переписуванні ядра системи.

Тож мова програмування Java у поєднанні з Android Studio є оптимальним вибором для розробки мобільного застосунку HabitFlow. Вона забезпечує стабільність, продуктивність, безпеку, масштабованість та повну сумісність із екосистемою Android. Завдяки розвинутим інструментам розробки та широким можливостям Java дозволяє створити довговічний та гнучкий продукт, який може розвиватися разом із потребами користувачів.

1.4 Визначення сфери застосування розроблювальної системи

У сучасному світі все більше людей прагнуть покращити якість свого життя, впроваджуючи корисні звички, оптимізуючи власний час та підвищуючи рівень самодисципліни. Однак на практиці цей процес часто стикається з низкою труднощів: нестачею мотивації, відсутністю структурованого підходу, нерегулярністю виконання та брак зворотного зв'язку. Традиційні інструменти, такі як паперові щоденники або прості замітки в телефоні, перестають відповідати сучасним вимогам. Саме тому виникає потреба у створенні спеціалізованого, інтуїтивно зрозумілого інструменту, який би зробив процес формування звичок більш простим, мотивуючим та ефективним.

У контексті цих потреб рішенням є розробка мобільного додатку HabitFlow — системи, яка бере за основу ключові ідеї гейміфікації, сучасної поведінкової психології та UX-підходів. На відміну від існуючих аналогів, HabitFlow зосереджується не на кількості функцій, а на їх якості та зручності, пропонуючи мінімалістичний, ненавантажений інтерфейс, доступний для людей різного віку та рівня цифрової грамотності.

Основна ідея полягає у створенні мобільного додатку, який допомагатиме користувачу формувати корисні звички, використовуючи легкий та приємний інтерфейс, що не потребує глибоких технічних знань. Система дозволяє:

- створювати набір звичок, які користувач прагне розвивати щодня;
- відстежувати прогрес за допомогою візуальних елементів, включаючи рівні, досвід і зростання “віртуальних рослин” — елементів гейміфікації, що стимулюють регулярність;

- налаштовувати профіль та персоналізувати взаємодію через особистий кабінет;
- отримувати нагороди та досягнення, які підсилюють внутрішню мотивацію та підтримують інтерес до процесу трекінгу.

Однією з ключових особливостей системи є акцент на гейміфікації, яка виконує роль м'якого, ненав'язливого мотиваційного механізму. Відповідно, мобільний додаток не лише фіксує виконання звичок, а й створює для користувача відчуття прогресу, розвитку та "росту" – що значно підвищує рівень залученості та зменшує ймовірність відмови від нових поведінкових патернів.

Для забезпечення коректної роботи системи також передбачається наявність адмін-панелі, яка дозволить адміністраторам керувати контентом, оновлювати інформацію, додавати або редагувати довідкові матеріали, підтримувати порядок у базі користувачів та статистиці. Такий інструмент дозволяє гнучко розширювати систему, підтримувати її актуальність та забезпечувати стабільне функціонування. При цьому адмін-панель розробляється з розрахунком на те, що нею користуватимуться люди без спеціальної технічної підготовки — тому інтерфейс буде простим, логічним і зрозумілим.

Мінімалістичний дизайн, відсутність зайвих елементів, продумана структура та чітка логіка взаємодії роблять HabitFlow доступним для максимально широкої аудиторії — від школярів, які хочуть сформувати перші корисні звички, до дорослих, зайнятих людей, що прагнуть впорядкувати своє життя. Такий підхід сприяє більшій залученості, знижує бар'єри входу та робить додаток зручним у повсякденному використанні.

У підсумку, впровадження мобільного додатку HabitFlow дозволяє автоматизувати та оптимізувати процес формування та підтримки звичок, підвищити рівень мотивації користувачів, а також створити ефективний інструмент для саморозвитку. Завдяки комбінації сучасних технологій, гейміфікаційних механік та простоти використання, система має потенціал стати корисним рішенням для широкої групи людей та ефективно впливати на їх повсякденну поведінку.

1.5 Призначення розробки

Метою даної розробки є створення сучасної системи для автоматизації процесу формування та відстеження корисних звичок. Основним призначенням

системи є надання користувачу зручного, інтуїтивно зрозумілого та мотивуючого інструмента, який спрощує щоденну роботу із саморозвитком і підвищує ймовірність довготривалого закріплення нових поведінкових патернів.

HabitFlow покликаний вирішувати проблему нестачі мотивації та відсутності структурованості під час формування звичок, використовуючи гейміфікаційні механізми, візуальні індикатори прогресу та ергономічний інтерфейс. Система має забезпечувати як зручність для кінцевого користувача, який працює зі своїми звичками, так і простоту адміністрування для модераторів та розробників, що підтримують роботу додатку.

Ключові складові розробки включають:

- Реляційну систему баз даних, призначену для зберігання структурованої інформації про користувачів, їх профілі, створені звички, історію виконання, прогрес, досягнення та інші важливі параметри, що формують повноцінну картину взаємодії з додатком.
- Інтерфейс доступу до даних у вигляді мобільного додатку, що надає користувачам можливість швидко створювати та редагувати звички, переглядати свою статистику, отримувати візуальні підказки та мотиваційні елементи. Мобільний інтерфейс також включає окрему панель для адміністраторів, яка дозволяє керувати глобальними параметрами системи, даними користувачів і структурою контенту.

Окрім цього, розробка має забезпечити стабільну, безпечну та масштабовану роботу системи, що дозволить використовувати її широкому колу користувачів без втрати швидкодії та доступності. Створення такої системи спрямоване на підвищення ефективності самоконтролю, розвиток самоорганізованості та підтримку позитивних змін у способі життя користувачів.

1.6 Мета створення системи

Метою створення системи є розробка сучасного мобільного додатку, який допомагає користувачам формувати корисні звички та підтримувати довгострокову мотивацію за рахунок елементів гейміфікації. Система спрямована на те, щоб зробити процес розвитку звичок більш цікавим, зрозумілим і приємним, водночас забезпечуючи користувача персоналізованим досвідом та наочним відстеженням прогресу.

З боку організації (розробника та потенційного сервісу) впровадження системи дозволяє:

- Надати новий інноваційний сервіс, орієнтований на покращення продуктивності та психологічного комфорту користувачів через механіки гейміфікації.
- Зменшити навантаження на консультативні чи супровідні сервіси, адже система автоматично допомагає користувачу відстежувати прогрес, формувати нагадування та аналізувати результати.
- Автоматизувати процеси збирання статистики щодо поведінки користувачів, виконання ними завдань та динаміки формування звичок.
- Забезпечити гнучке керування контентом — додавання нових звичок, рекомендацій, завдань та віртуальних винагород без необхідності складної ручної роботи.
- Залучити ширшу аудиторію за рахунок сучасного інтерфейсу та ігрових елементів, які підвищують загальну привабливість продукту.

З боку користувачів система створена для того, щоб:

- Значно спростити процес формування звичок, зробивши його структурованим, зрозумілим і приємним завдяки візуалізаціям, нагородам та ігровим механікам.
- Зменшити психологічний бар'єр, який часто виникає під час початку нової діяльності, адже гейміфікація сприяє легкому входженню у процес.
- Скоротити час на пошук корисної інформації та планування — користувач отримує вже готові звички, рекомендації, трекінг та нагадування.
- Підвищити рівень мотивації завдяки віртуальним нагородам, прогрес-барам, досягненням, зростанню «віртуальних рослин» чи інших візуальних індикаторів, які допомагають користувачу бачити свій результат.
- Забезпечити комфортний доступ до системи з будь-якого пристрою, отримуючи постійний контроль над власними цілями та досягненнями.

У підсумку система спрямована на створення цифрового інструмента, який підтримує користувача в його щоденних зусиллях, підвищує рівень самодисципліни та робить процес формування звичок більш ефективним, цікавим та приємним.

1.7 Основні функціональні можливості системи

Мобільний додаток HabitFlow забезпечує широкий спектр функціональних можливостей, спрямованих на підтримку користувача в формуванні, відстеженні та покращенні щоденних звичок. Система розроблена таким чином, щоб забезпечити максимально інтуїтивну взаємодію, враховуючи потреби сучасних користувачів, які очікують швидку, доступну та персоналізовану роботу застосунку. Додаток створений в Android Studio та орієнтований на використання на мобільних пристроях з операційною системою Android, що дозволяє пропонувати оптимізований досвід із застосуванням сучасних технологій Android SDK.

Функціональні можливості для звичайного користувача

Користувач, встановивши додаток та створивши власний обліковий запис, отримує можливість працювати з повноцінним інструментом для управління своїми звичками. Основні можливості включають:

- Створення та налаштування звичок. Користувач може додавати нові звички, задаючи їх назву, короткий опис, періодичність виконання, бажаний час нагадувань та інші ключові параметри. Завдяки гнучкості інтерфейсу, кожна звичка може бути персоналізована під індивідуальні потреби.
- Перегляд списку активних звичок. Додаток забезпечує зручний візуальний список усіх звичок користувача, що дозволяє швидко оцінити загальну кількість поточних активностей, їх статус та прогрес виконання.
- Відзначення виконання звички. HabitFlow дозволяє щоденно фіксувати виконання чи пропуск звички, що автоматично відображається в системі статистики та прогресу. Цей механізм формує в користувача відчуття досягнення та мотивації.
- Налаштування нагадувань. Користувач може створювати один або кілька нагадувань для кожної звички. Додаток використовує можливості Android для планування сповіщень, що гарантує своєчасне нагадування навіть у фоновому режимі.
- Перегляд статистики та прогресу. Система надає зручні графічні та аналітичні відомості про ефективність виконання звичок: щотижневі та щомісячні графіки, відсоток виконання, серії успішних днів тощо. Це дозволяє користувачу бачити свій прогрес та мотивує продовжувати роботу над звичками.

- Редагування та видалення звичок. У будь-який момент користувач може змінити параметри звички або повністю видалити її, якщо вона більше не є актуальною.
- Робота з профілем користувача. Доступ до особистого кабінету дозволяє переглядати та змінювати інформацію профілю (ім'я, електронну пошту, аватар), керувати налаштуваннями та конфіденційністю.

Функціональні можливості для адміністратора системи

Для забезпечення якісної підтримки додатка та оновлення контенту передбачена адміністративна частина, яка доступна лише користувачам із відповідними правами:

- Управління глобальними параметрами системи. Адміністратор може змінювати загальні налаштування додатку, наприклад, дефолтні параметри для нових звичок, частоту синхронізації чи правила збереження даних.
- Контроль контенту. Адміністратор має можливість оновлювати довідкову інформацію, змінювати текстові блоки, додавати нові рекомендації або корисні поради щодо формування звичок.
- Управління користувачами. При необхідності адміністратор може виконувати функції модерації: переглядати статистику користувача, блокувати облікові записи або відновлювати доступ у разі технічних проблем.
- Перегляд аналітики та загальної статистики додатку. Адміністратор має доступ до агрегованих даних про активність додатка — кількість нових користувачів, популярність окремих функцій, частоту взаємодії зі сповіщеннями. Це сприяє подальшому вдосконаленню системи.

2 ДОСЛІДЖЕННЯ ІСНУЮЧИХ ПІДХОДІВ, МЕТОДІВ, ЗАСОБІВ, ТЕХНОЛОГІЙ, ІНСТРУМЕНТАРІЮ ДЛЯ ВИРІШЕННЯ ПОДІБНИХ ЗАДАЧ

2.1 Огляд наукових підходів до формування звичок

Проблема формування та підтримки корисних звичок є предметом дослідження в психології, когнітивних науках та інформаційних технологіях. У роботах Ч. Дахіга (*Duhigg, 2012*) та Б. Фогга (*Fogg, 2009*) описано моделі поведінкових змін, які стали базою для сучасних мобільних застосунків самоменеджменту. Модель *cue–routine–reward* (сигнал–рутина–нагорода) визначає, що звичка формується шляхом повторення певної дії під впливом стимулу з наступним підкріпленням.

Дослідження *Wood & Neal (2007)* довели, що автоматизація поведінки можлива лише за умов стабільного контексту виконання дії, а тому мобільні застосунки мають не лише фіксувати активність користувача, а й створювати контекст для повторення поведінки — наприклад, через нагадування, візуальні підказки чи елементи гри.

2.2 Гейміфікація як інструмент мотивації

Гейміфікація як метод підвищення залученості користувачів широко досліджується у працях *Werbach & Hunter (2012)*, *Hamari, Koivisto & Sarsa (2014)*. У цих дослідженнях показано, що впровадження ігрових механік у неігрові системи суттєво покращує мотивацію користувачів та ефективність виконання завдань.

У роботі *Sailer et al. (2017)* доведено, що найбільш ефективними є ті елементи гейміфікації, які задовольняють базові психологічні потреби:

- Автономію (можливість обирати власні цілі);
- Компетентність (відчуття прогресу через рівні, бали, досягнення);
- Соціальну взаємодію (змагання, співпраця, лідерборди).

Дослідження *Johnson et al. (2016)* у сфері мобільних застосунків для формування звичок показало, що гейміфіковані елементи, такі як візуальні нагороди, серії виконань (“*streaks*”) та персональні аватари, сприяють тривалому утриманню користувачів у системі на 30–40% довше, ніж у звичайних трекерах.

2.3 Використання гейміфікації в мобільних системах підтримки здоров'я

Зарубіжні дослідження підтверджують ефективність гейміфікації не лише у сфері навчання, а й у контексті формування здорових звичок. Наприклад, у роботі *Edwards et al. (2016)* розглянуто вплив гейміфікаційних елементів на фізичну активність користувачів через застосунок “Fitocracy”. Автори виявили, що система рівнів і балів викликає значне зростання активності в перші 4 тижні використання.

Дослідження *Cugelman (2013)* систематизувало принципи цифрової поведінкової психології, виділивши сім основних механізмів, серед яких соціальне підкріплення, зворотний зв'язок і самомоніторинг. Саме ці принципи лежать в основі більшості сучасних мобільних застосунків для формування звичок.

У роботі *Orji & Moffatt (2018)* акцентовано увагу на необхідності персоналізації гейміфікації — тобто адаптації ігрових елементів до типу користувача. Це особливо актуально для довготривалих систем, таких як застосунки для підтримки здоров'я чи саморозвитку.

2.4 Технологічні підходи у реалізації систем формування звичок

У технічному аспекті більшість рішень базуються на клієнт–серверній архітектурі з використанням REST API. Як зазначено в *Arias et al. (2020)*, інтеграція гейміфікаційних елементів можлива через окремі модулі бекенду, що дозволяє гнучко змінювати правила гри без оновлення клієнтського додатку.

Робота *Jain et al. (2021)* підкреслює роль аналітичних модулів у таких системах, які збирають статистику дій користувачів та на основі машинного навчання прогнозують ризик втрати мотивації, пропонуючи відповідні рекомендації.

2.5 Висновки дослідження

Огляд наукових джерел показує, що використання гейміфікації у системах формування звичок має доведену ефективність, особливо при поєднанні з поведінковими моделями та персоналізацією під користувача. Однак більшість досліджень зосереджені або на психологічному, або на технічному аспекті,

залишаючи відкритим питання інтеграції комплексної моделі — тобто поєднання гейміфікаційних механік, аналітики даних та гнучкої серверної архітектури. Саме це завдання і вирішується у даному проєкті — створення мобільного застосунку, який поєднує методи гейміфікації, психологічні моделі формування звичок і сучасні технології обробки даних.

3 ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

Метою даного дослідження є проектування та реалізація мобільного застосунку для підтримки формування корисних звичок із використанням елементів гейміфікації, що сприяють підвищенню мотивації користувачів та зручності взаємодії із системою.

Завдання дослідження полягає у створенні технологічного рішення, яке забезпечить ефективне управління даними користувачів, відстеження прогресу у формуванні звичок та інтеграцію ігрових механік у процес саморозвитку.

Основні етапи реалізації технології

1. Аналіз вимог та постановка функціональних задач.

На цьому етапі проводиться визначення цільової аудиторії застосунку, формування вимог до функціональності (створення звичок, щоденне відстеження виконання, нагороди, рівні тощо) та визначення нефункціональних вимог (зручність інтерфейсу, надійність збереження даних, продуктивність).

2. Проектування архітектури системи.

Розробка архітектури, що передбачає:

- серверну частину, реалізовану на Java у середовищі IntelliJ IDEA з використанням Spring Framework;
- клієнтську частину, створену у Android Studio на Kotlin або Java;
- базу даних MySQL, що використовується для зберігання інформації про користувачів, їхні звички, рівень досягнень, нагороди тощо.

3. Вибір технологій та інструментів.

Для реалізації бекенду використовується стек технологій:

- Spring Boot — для створення REST API;
- JPA/Hibernate — для роботи з базою даних;
- MySQL — для збереження структурованих даних.

Для фронтенду:

- Android SDK — для реалізації інтерфейсу та взаємодії з сервером;
- Material Design — для забезпечення інтуїтивного UX/UI.

4. Розробка функціональних модулів.

Модулі застосунку поділяються на:

- Модуль аутентифікації користувачів (реєстрація, вхід, збереження сесії);

- Модуль управління звичками (створення, редагування, відстеження прогресу);
- Модуль гейміфікації (досягнення, рівні, винагороди);
- Модуль аналітики (перегляд статистики, візуалізація прогресу).

5. Тестування та оптимізація.

Передбачено проведення функціонального тестування (перевірка кожного модуля окремо) та інтеграційного тестування (перевірка взаємодії між сервером, клієнтом і базою даних).

6. Розгортання та оцінка результатів.

Після реалізації застосунків буде протестовано на групі користувачів для оцінки ефективності гейміфікаційних механік у підвищенні рівня мотивації до формування звичок.

4 РОЗРОБКА МЕТОДУ (ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ) ВИРІШЕННЯ ЗАДАЧІ

4.1 Розробка системних вимог до системи

У процесі створення мобільного застосунку HabitFlow, основним призначенням якого є допомога користувачам у формуванні та підтриманні корисних звичок за рахунок сучасних методів гейміфікації, важливо чітко визначити системні вимоги, що забезпечать стабільну роботу, зручність використання та можливість подальшого розвитку застосунку. Нижче наведено орієнтовний перелік системних вимог, які формують основу для коректного функціонування мобільного рішення.

Операційна система

Мобільний застосунок HabitFlow повинен стабільно працювати на сучасних мобільних пристроях під керуванням операційної системи Android. Мінімальна підтримувана версія Android має бути визначена таким чином, щоб забезпечити баланс між широким охопленням аудиторії та доступом до сучасних API, що дозволяють реалізувати візуальні ефекти, анімації, push-сповіщення та інші функції, необхідні для якісної взаємодії з користувачем. Рекомендовано забезпечити підтримку Android 10 (API 29) та новіших версій.

Середовище розробки

Розробка здійснюється у середовищі Android Studio, яке є офіційним інструментом створення Android-додатків та надає широкий набір можливостей: від зручного інтерфейсу дизайну екранів до інструментів відлагодження, тестування й оптимізації. Завдяки інтеграції з Gradle забезпечується гнучке керування залежностями, модулями та збіркою проєкту.

Мова програмування та фреймворки

Для реалізації функціонального ядра застосунку використовується Kotlin як сучасна, безпечна та офіційно рекомендована Google мова програмування для Android-розробки. Kotlin забезпечує лаконічний синтаксис, високу продуктивність та вбудовану підтримку об'єктно-орієнтованого та функціонального стилів програмування.

Для побудови інтерфейсу можуть застосовуватися:

- XML Layouts або Jetpack Compose — залежно від обраної архітектури та сучасності підходу.

- Модулі Android Jetpack (ViewModel, LiveData або StateFlow, Navigation Component), що полегшують масштабування та підтримку чистої архітектури.

База даних

Мобільний застосунок повинен мати локальну базу даних для зберігання інформації про звички, прогрес, нагороди користувача та інші пов'язані дані. Для цього рекомендується використання:

- Room Database, що забезпечує безпечну роботу з локальними даними, спираючись на SQLite;
- або DataStore для зберігання легких налаштувань, станів і конфігурацій застосунку.

Це дозволить працювати навіть у режимі офлайн та забезпечить зручний доступ до інформації без постійного підключення до мережі.

Користувацький інтерфейс

Інтерфейс HabitFlow повинен бути інтуїтивно зрозумілим, візуально приємним і адаптованим під екрани різних розмірів. Оскільки застосунок взаємодіє з користувачем щодня, важливо забезпечити:

- просту навігацію між екранами,
- м'які анімації та плавні переходи,
- елементи гейміфікації (рослини, прогрес-бари, віртуальні нагороди тощо),
- адаптивну типографіку та кольорову палітру, що сприяє позитивному користувацькому досвіду.

Безпека

Для мобільного застосунку, що працює з персональними даними користувача, важливо гарантувати базові засоби безпеки. Вимоги включають:

- ізоляцію локально збережених даних,
- захист від несанкціонованого доступу,
- контроль прав доступу до системних функцій (наприклад, push-сповіщень),
- можливе використання Firebase Authentication у випадку додавання хмарної синхронізації.

Продуктивність

HabitFlow має працювати швидко та стабільно, не перевантажуючи оперативну пам'ять та не спричиняючи надмірного споживання батареї. Важливо забезпечити:

- оптимізоване завантаження ресурсів,

- ефективно управління потоками,
- мінімальне використання важких анімацій,
- швидку обробку користувацьких дій.

Надійність та розширюваність

При проєктуванні системи передбачається можливість майбутнього розширення — наприклад, додавання соціальних функцій, інтеграції з хмарними сервісами, системи нагород, додаткових типів гейміфікації тощо. Архітектура застосунку має бути модульною, легко адаптованою до змін і доповнень.

Інструменти супроводу та розробки

Для підвищення якості коду та стабільності системи рекомендується використовувати:

- Git для контролю версій,
- Firebase Crashlytics або аналогічні сервіси для відстеження помилок,
- JUnit / Espresso для автоматизованого тестування,
- статичний аналізатор коду (наприклад, Detekt).

4.2 Розробка функціональної моделі

Використовуючи стандарт IDEF0, вдалося створити функціональну модель, яка відображає структуроване зображення функцій інформаційної системи, а також інформацію та об'єкти, що зв'язують ці функції.

На рисунку 4.1 відображено функціональну модель «Формування корисних звичок користувача».

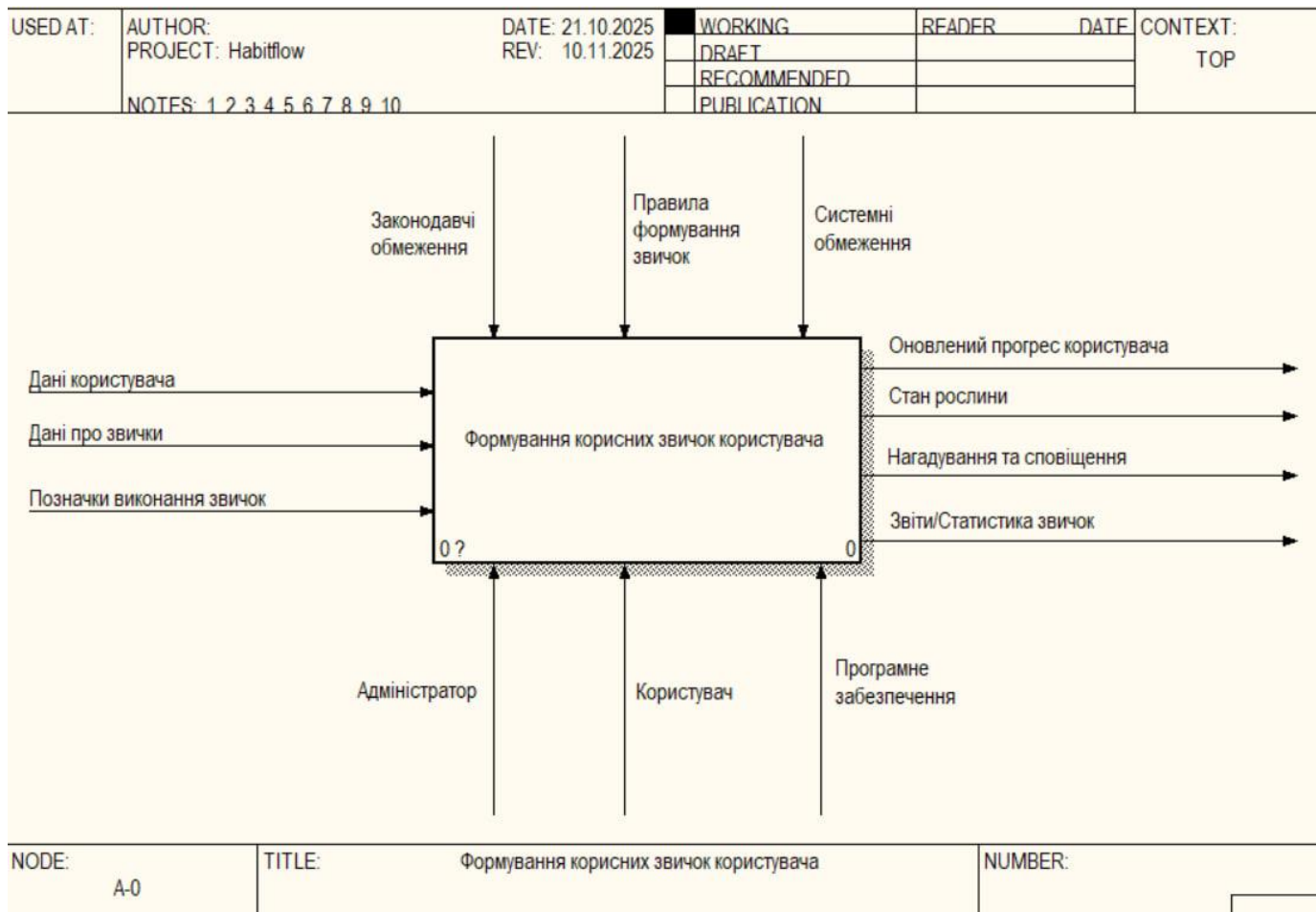


Рисунок 4.1 – Функціональна модель «Формування корисних звичок користувача»

Метою розробки моделі є визначення функціональних вимог до ІС. Розроблена система має два види користувачів (адміністратор та користувач). Ліворуч від функціонального блоку знаходяться вхідні дані, необхідні для виконання функцій:

- Дані користувача
- Дані про звички
- Позначки виконання звичок

Над функціональним блоком є дуги, які встановлюють деякі правила, стратегії або стандарти, які впливають на продуктивність функції:

- Законодавчі обмеження
- Системні обмеження
- Правила формування звичок

Праворуч від функціонального блоку зображені дуги вихідних даних:

- Оновлений прогрес користувача
- Стан рослини
- Нагадування та сповіщення

- Звіти/Статистика звичок

Для більш детального представлення була проведена декомпозиція контекстної діаграми.

На рисунку 4.2 представлена діаграма декомпозиції.

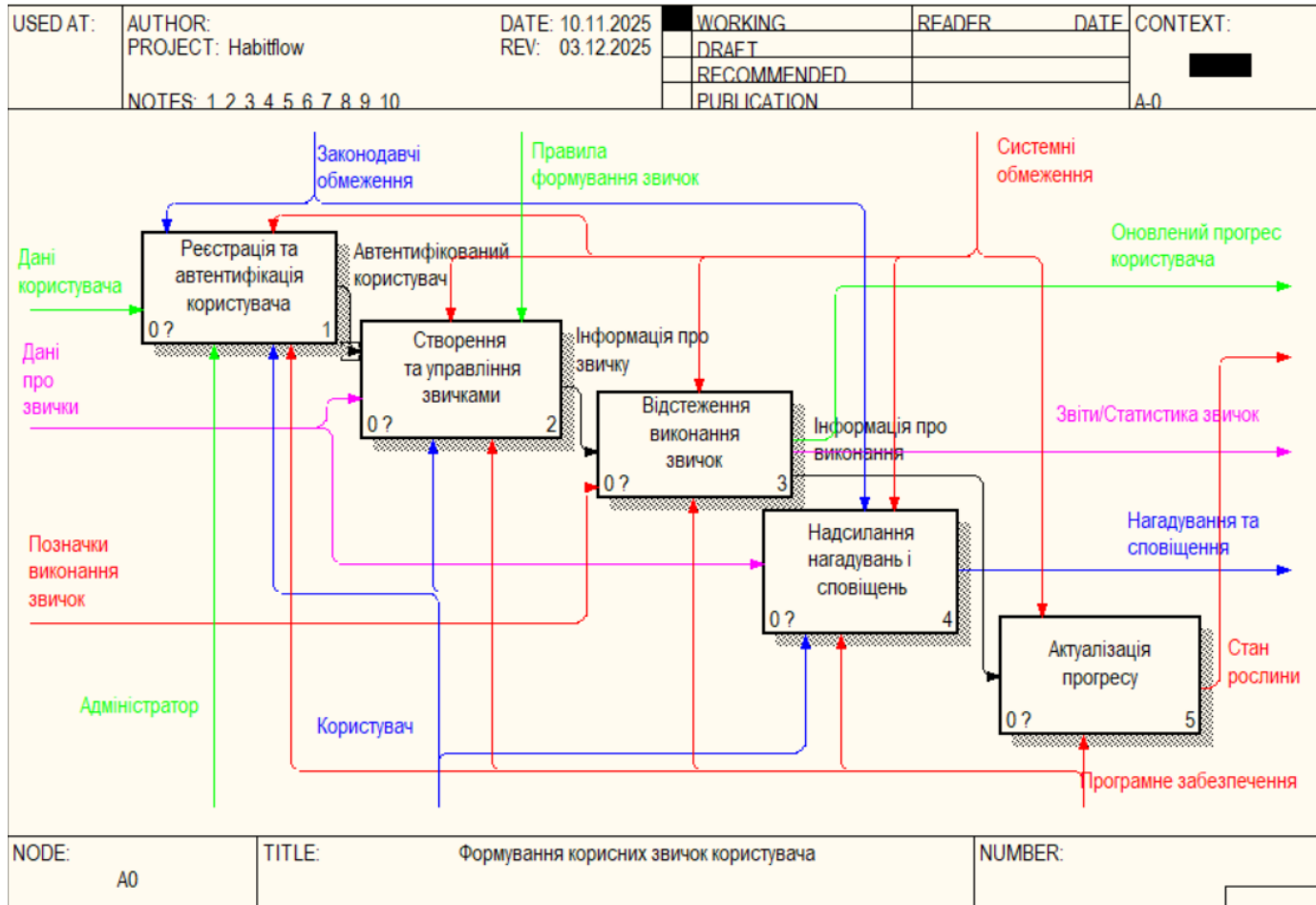


Рисунок 4.2 – Декомпозиція контекстної діаграми

Вона містить п'ять основних бізнес-процесів і всі дуги, які описуються на етапі побудови концептуальної моделі (Рисунок 4.1), а саме:

- процес реєстрації та автентифікації користувача приймає вхідні дані від користувача, такі як ім'я, електронна пошта, пароль та інші параметри, необхідні для створення або перевірки облікового запису. У результаті процесу користувач проходить автентифікацію, отримує доступ до персонального профілю та своїх даних про звички. Процес контролюється за допомогою законодавчих обмежень (захист персональних даних, конфіденційність, політика безпеки) та системних обмежень (формати введених даних). Ресурси, за допомогою яких виконується робота, включають користувача, адміністратора та програмне забезпечення системи.

- процес створення та управління звичками приймає від користувача дані про нову звичку — її назву, ціль, частоту виконання, тривалість тощо. Як наслідок, система створює або оновлює запис про звичку, зберігаючи відповідну інформацію у базі даних. Процес контролюється правилами формування звичок (структура даних, логіка повторення, допустимі параметри) та системними обмеженнями (кількість звичок, формат введених даних). Ресурсами, які забезпечують виконання процесу, є користувач та програмне забезпечення.

- процес відстеження виконання звичок отримує від користувача позначки про виконання конкретних звичок у визначений час або день. У результаті система оновлює прогрес користувача, розраховує показники регулярності та формує статистичні дані про ефективність виконання. Процес контролюється системними обмеженнями (допустимі значення, частота оновлення). Ресурси, які забезпечують виконання процесу, включають програмне забезпечення.

- процес надсилання нагадувань і сповіщень використовує дані про розклад звичок і активність користувача, щоб формувати своєчасні повідомлення про необхідність виконати звичку. Як результат, користувач отримує push-сповіщення або внутрішні повідомлення, які підвищують мотивацію та підтримують регулярність дій. Процес контролюється системними обмеженнями (розклад, ресурси пристрою) та законодавчими обмеженнями (дотримання вимог щодо обробки персональних даних, конфіденційності та електронних повідомлень). Ресурси, що забезпечують виконання процесу, включають програмне забезпечення та користувача.

- процес актуалізації прогресу приймає дані про виконання звичок та рівень прогресу користувача, інтерпретуючи їх у вигляді візуальної метафори — росту віртуальної рослини. У результаті система відображає оновлений стан рослини, який символізує розвиток користувача, а також формує статистику звичок і звіти. Процес контролюється системними обмеженнями (дизайн і межі рівнів). Ресурси процесу включають програмне забезпечення, яке здійснює обчислення й оновлення UI.

Для того, щоб детальніше розглянути бізнес-процеси, було вирішено провести декомпозицію для кожного з них.

На рисунку 4.3 відображено декомпозицію бізнес-процесу «Реєстрація та автентифікація користувача».

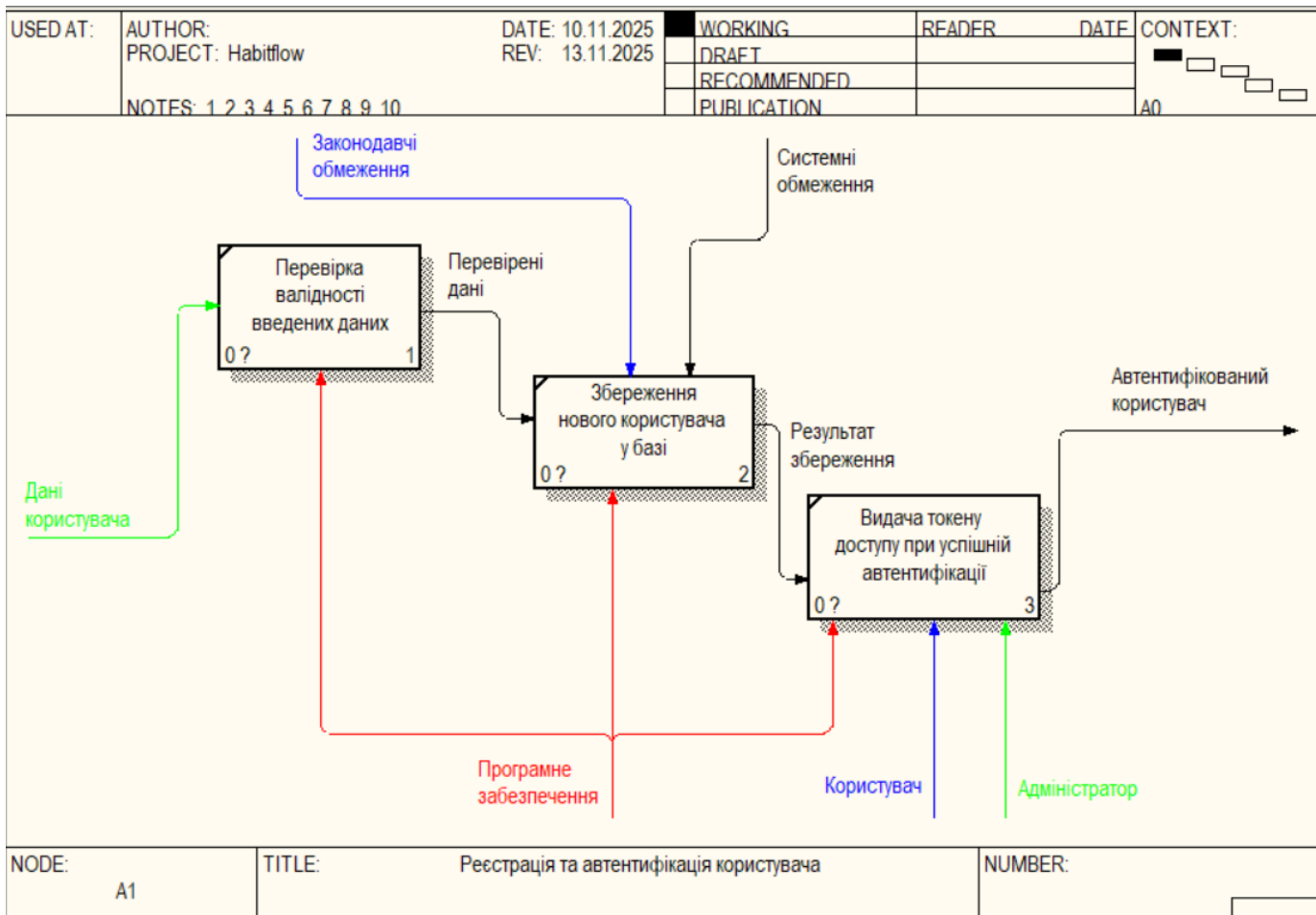


Рисунок 4.3 – Декомпозиція бізнес-процесу «Реєстрація та автентифікація користувача»

Бізнес-процес було поділено на три бізнес-функції:

- функція перевірки валідності введених даних відповідає за контроль правильності та повноти інформації, яку користувач вводить під час реєстрації або входу в систему. Програмне забезпечення перевіряє коректність полів (наприклад, формат електронної пошти, мінімальну довжину пароля, відсутність порожніх значень).

Ресурси, за допомогою яких виконується робота, включають програмне забезпечення.

- функція збереження нового користувача у базі відповідає за запис перевірених даних нового користувача до системи. Після успішної перевірки дані передаються до блоку програмного забезпечення, який створює новий запис у базі користувачів. Функція контролюється законодавчими та системними обмеженнями.

Ресурси, за допомогою яких виконується робота, включають програмне забезпечення.

- функція видачі токена доступу реалізує процес створення унікального токена після успішної перевірки облікових даних користувача. Цей токен підтверджує особу користувача та дозволяє йому здійснювати подальші дії у системі без повторного введення логіну та пароля.

Ресурси, за допомогою яких виконується робота, включають програмне забезпечення, користувача та адміністратора додатку.

На рисунку 4.4 відображено декомпозицію бізнес-процесу «Створення та управління звичками».

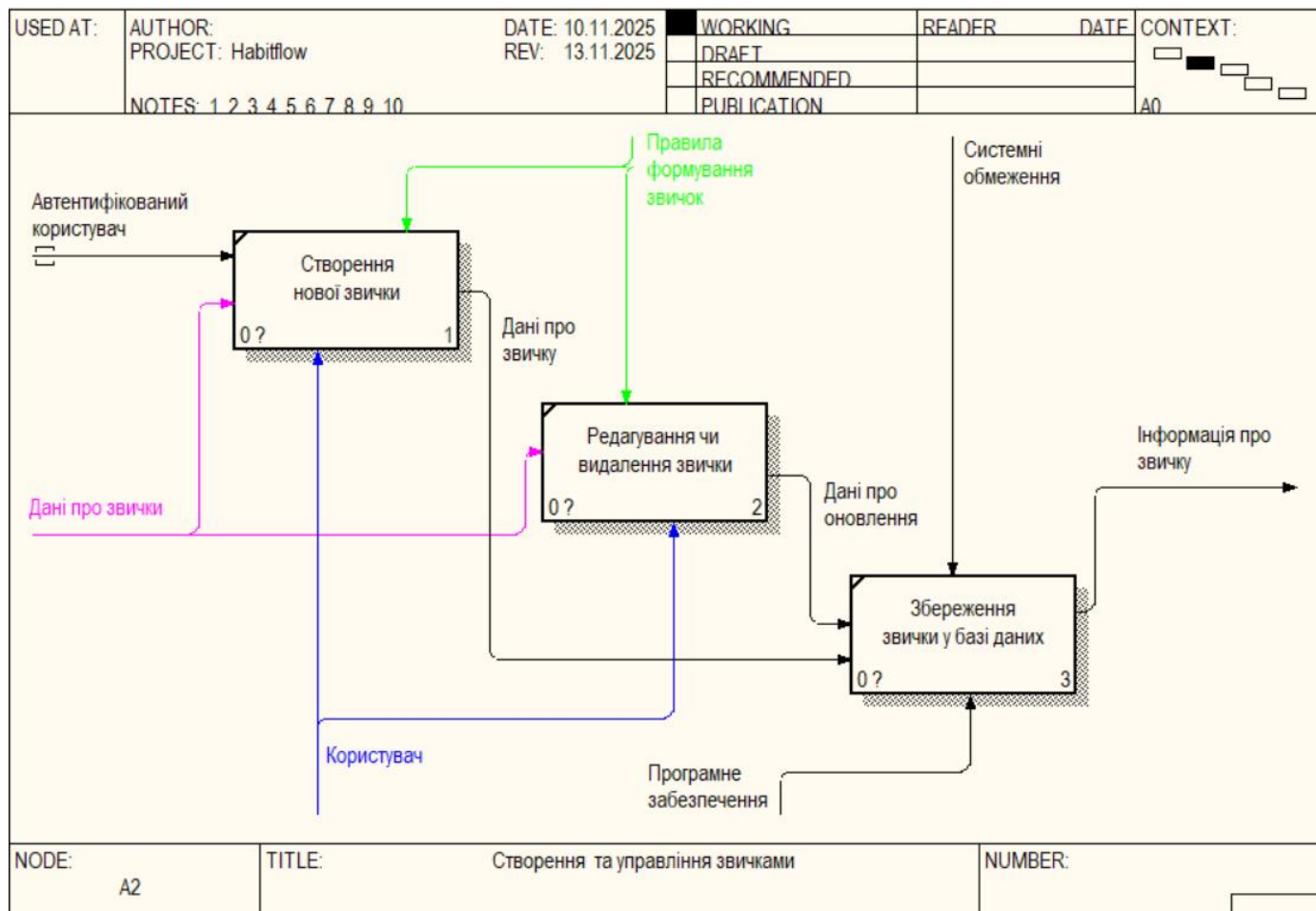


Рисунок 4.4 – Декомпозиція бізнес-процесу «Створення та управління звичками»

Бізнес-процес було поділено на три бізнес-функції:

- функція створення нової звички відповідає за формування нової звички користувачем у системі. Користувач вводить параметри звички (назву, опис, періодичність, нагадування тощо), після чого система перевіряє їх відповідність правилам формування звичок. Функція контролюється правилами формування звички, які визначають структуру, обов'язкові поля та можливі обмеження.

Ресурс, за допомогою якого виконується робота, — користувач.

- функція редагування чи видалення звички забезпечує можливість користувачу змінювати параметри вже створених звичок або видаляти їх із системи. Користувач ініціює зміни через інтерфейс, після чого програмне забезпечення перевіряє коректність дій та вносить зміни. Функція контролюється правилами формування звички, які визначають, які зміни допустимі, а також послідовність дій при видаленні.

Ресурс, за допомогою якого виконується робота, — користувач.

- функція збереження звички у базі даних відповідає за фіксацію створених або змінених звичок у системі. Після підтвердження дій користувача система записує дані до сховища, забезпечуючи цілісність і доступність інформації для подальшого відстеження прогресу. Функція контролюється системними обмеженнями, які визначають формат збереження, унікальність даних і порядок оновлення записів.

Ресурс, за допомогою якого виконується робота, — програмне забезпечення.

На рисунку 4.5 відображено декомпозицію бізнес-процесу «Відстеження виконання звички».

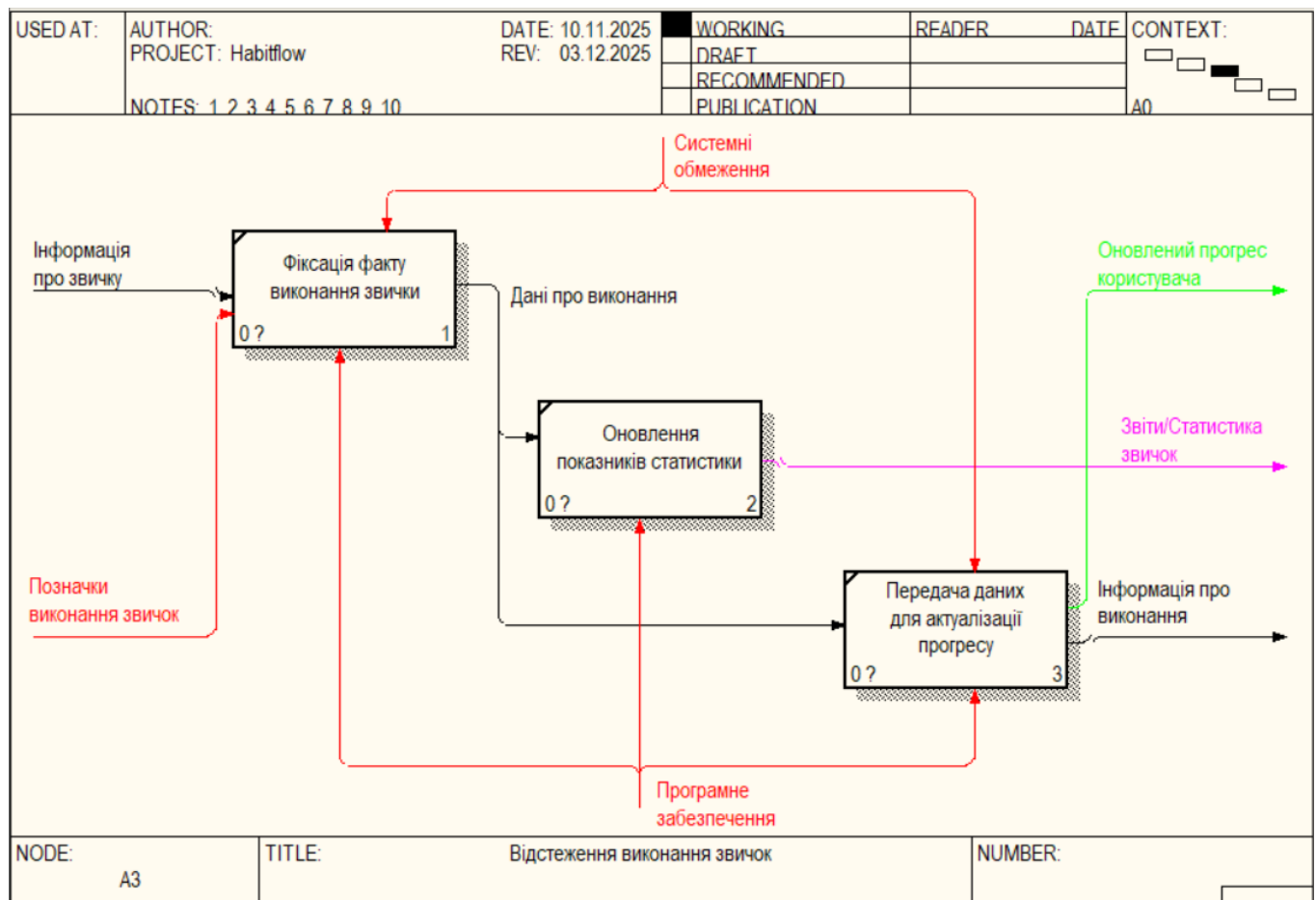


Рисунок 4.5 – Декомпозиція бізнес-процесу «Відстеження виконання звички»

Бізнес-процес було поділено на три бізнес-функції:

- функція фіксації факту виконання звички відповідає за реєстрацію дій користувача, пов'язаних із виконанням певної звички. Коли користувач позначає звичку як виконану, система приймає ці дані та зберігає запис про виконання у базі даних. Функція контролюється системними обмеженнями, які визначають правильність структури даних, унікальність записів і час фіксації.

Ресурс, за допомогою якого виконується робота, — програмне забезпечення.

- функція оновлення показників статистики відповідає за відображення динаміки виконання звичок користувача. На основі зафіксованих даних користувач може переглядати власний прогрес, відсоток успішності, серії виконань тощо.

Ресурс, за допомогою якого виконується робота, — користувач.

- функція передачі даних для актуалізації процесу відповідає за взаємодію між модулем звичок і модулем візуалізації прогресу через стан рослини. Система аналізує зміни у прогресі користувача та передає ці дані до підсистеми, що відображає стан рослини (наприклад, зростання чи зміна кольору). Функція контролюється системними обмеженнями, які визначають формат і періодичність передачі даних.

Ресурс, за допомогою якого виконується робота, — програмне забезпечення.

На рисунку 4.6 відображено декомпозицію бізнес-процесу «Надсилання нагадувань та сповіщень».

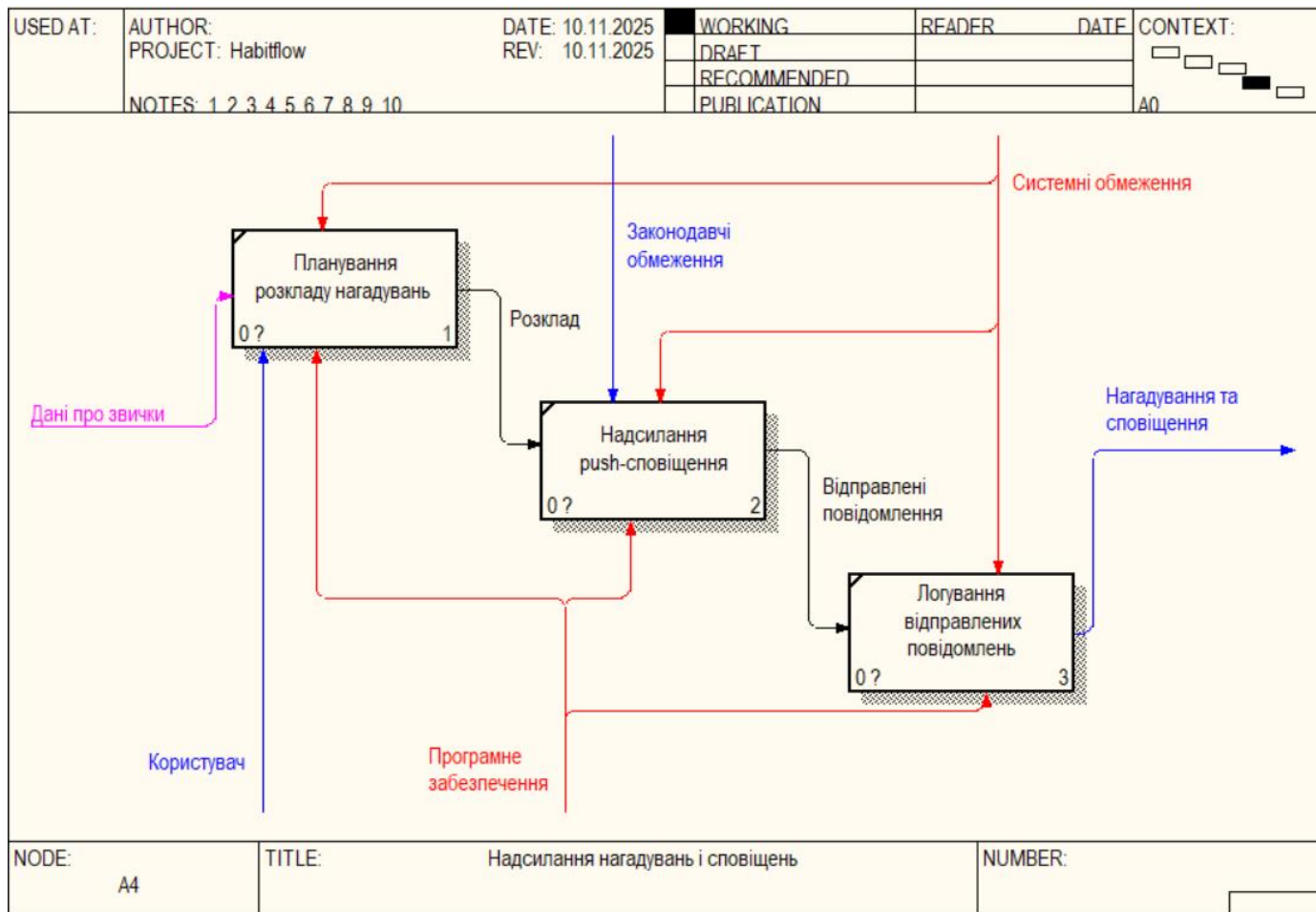


Рисунок 4.6 – Декомпозиція бізнес-процесу «Надсилання нагадувань та сповіщень»

Бізнес-процес було поділено на три бізнес-функції:

- функція планування розкладу нагадувань відповідає за визначення часу, частоти та умов надсилання повідомлень користувачеві. Користувач обирає бажаний час або періодичність отримання нагадувань, після чого система автоматично формує відповідний розклад. Функція контролюється системними обмеженнями, що визначають допустимі інтервали та формат збереження даних, а також передбачає участь користувача у налаштуванні параметрів.

Ресурси, за допомогою яких виконується робота, — програмне забезпечення та користувач.

- функція надсилання push-сповіщення відповідає за передачу користувачеві повідомлень про заплановані дії або невиконані звички. Система ініціює надсилання повідомлень відповідно до раніше сформованого розкладу нагадувань. Функція контролюється законодавчими (щодо обробки персональних даних і згоди на сповіщення) та системними обмеженнями (формат, частота та технічні умови надсилання).

Ресурс, за допомогою якого виконується робота, — програмне забезпечення.

- функція логування відправлених повідомлень відповідає за збереження інформації про факт надсилання нагадувань та сповіщень. Система фіксує час, тип повідомлення, статус доставки та інші технічні параметри, що можуть бути використані для аналітики або усунення збоїв. Функція контролюється системними обмеженнями, які визначають формат збереження логів та період їх оновлення.

Ресурс, за допомогою якого виконується робота, — програмне забезпечення.

На рисунку 4.7 відображено декомпозицію бізнес-процесу «Актуалізація процесу»

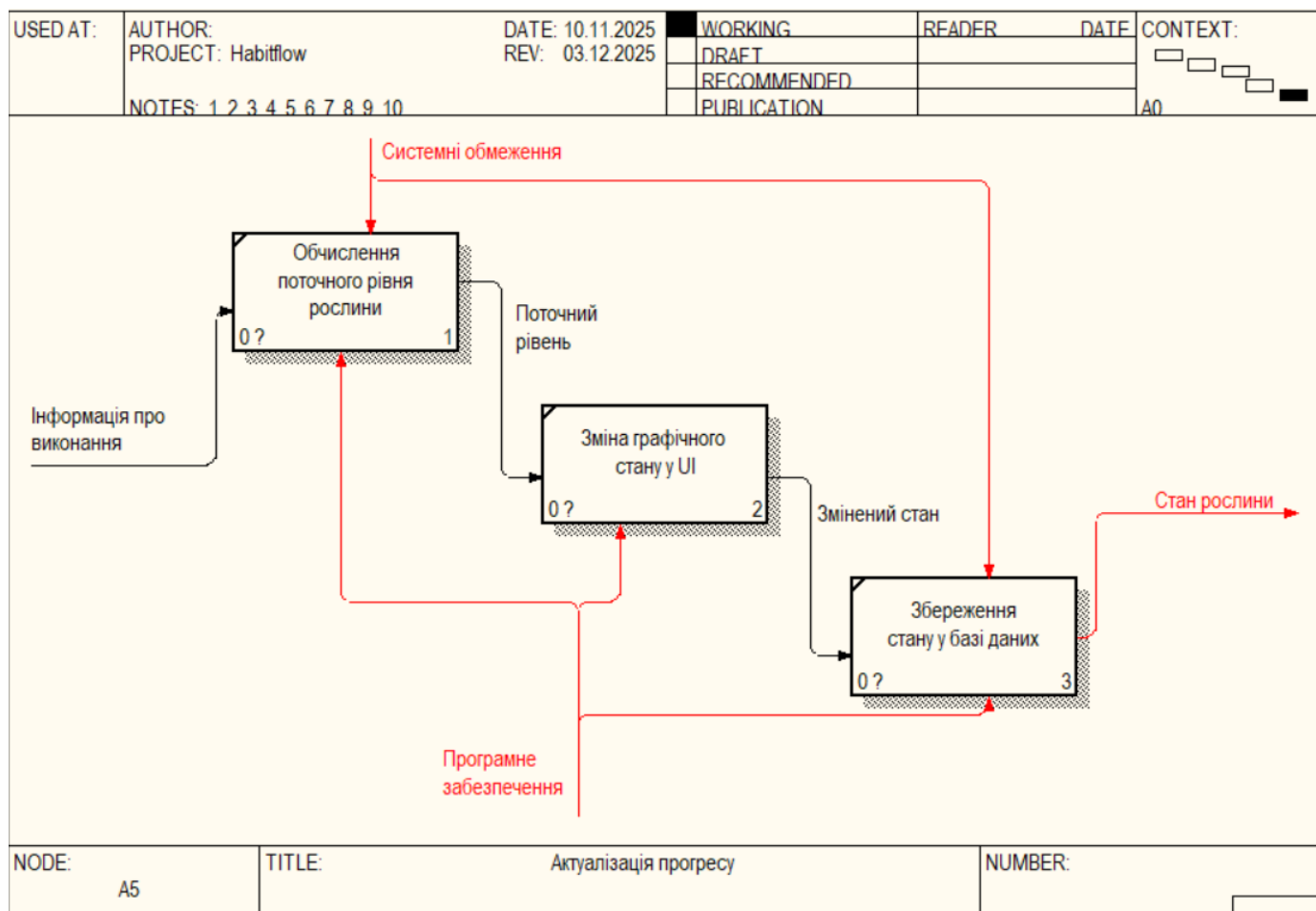


Рисунок 4.7 – Декомпозиція бізнес-процесу «Актуалізація процесу»

Бізнес-процес було поділено на три бізнес-функції:

- функція обчислення поточного рівня рослини відповідає за визначення рівня розвитку або стану віртуальної рослини відповідно до прогресу користувача у виконанні звичок. Система аналізує накопичені дані про виконання

завдань, враховує частоту, стабільність та успішність користувача, після чого обчислює показники для оновлення візуального стану рослини. Функція контролюється системними обмеженнями, що визначають алгоритм розрахунку рівня та допустимі параметри зміни стану.

Ресурс, за допомогою якого виконується робота, — програмне забезпечення.

- функція зміни графічного стану у UI відповідає за візуальне оновлення відображення стану рослини в інтерфейсі користувача. Після отримання результатів розрахунку система оновлює графічні елементи (наприклад, змінює вигляд або етап росту рослини), відображаючи досягнення користувача у зручній та мотиваційній формі. Функція не контролюється, оскільки не передбачає зовнішніх чи нормативних обмежень, а є частиною інтерфейсної логіки.

Ресурс, за допомогою якого виконується робота, — програмне забезпечення.

- функція збереження стану у базі даних відповідає за фіксацію оновленого рівня та стану рослини для забезпечення цілісності даних та можливості подальшого відновлення прогресу користувача. Система записує новий стан до бази даних, що дозволяє відтворювати актуальну інформацію при повторному вході в додаток або на інших пристроях користувача. Функція контролюється системними обмеженнями, які регламентують формат збереження, частоту оновлення та забезпечення узгодженості даних.

Ресурс, за допомогою якого виконується робота, — програмне забезпечення.

Для відображення ієрархії декомпозиції від глобальних функцій системи (Рисунок 4.1) до всіх її підфункцій нижнього рівня створено «Діаграму дерева вузлів» (Рисунок 4.8).

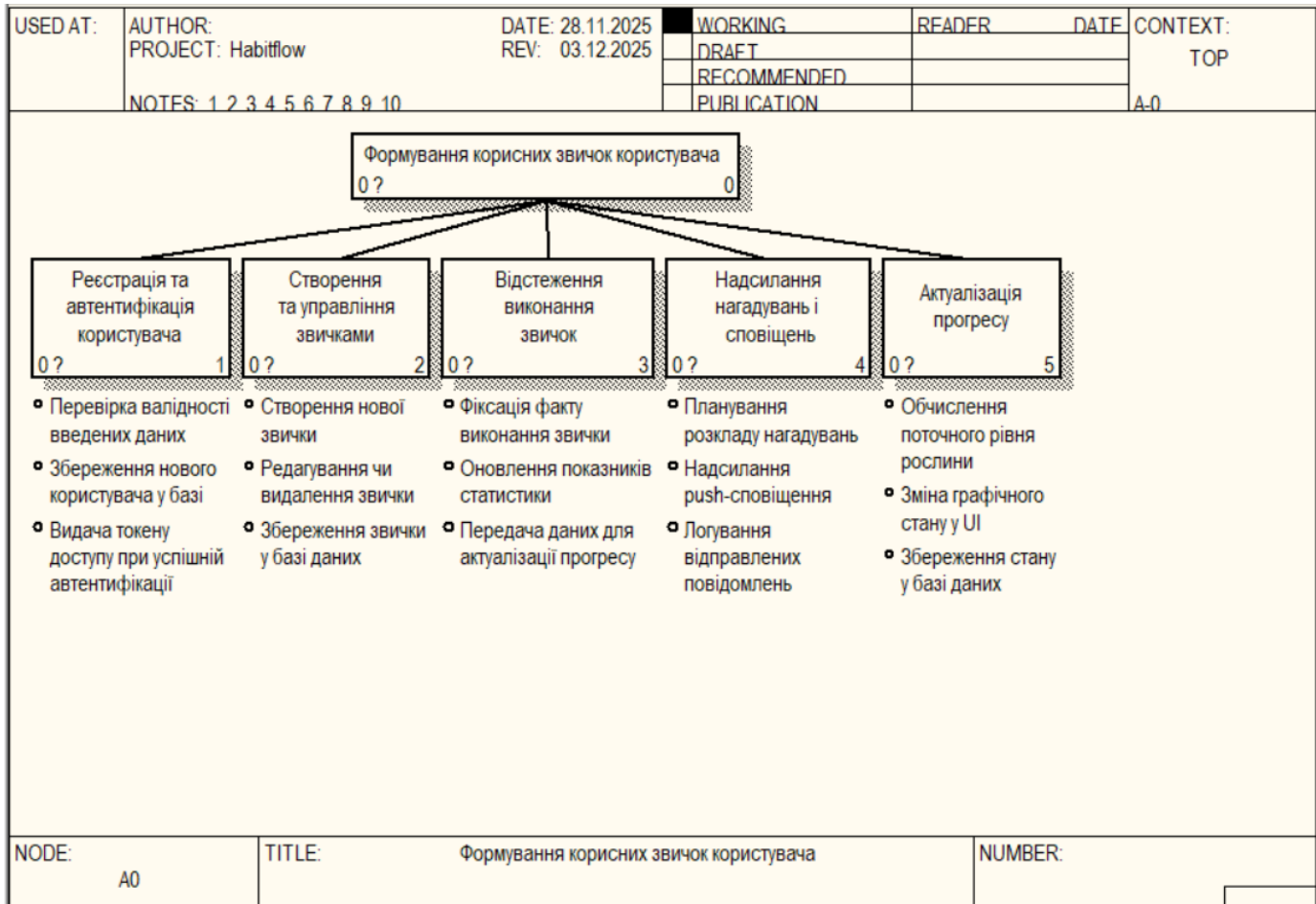


Рисунок 4.8 - Діграма дерева вузлів «Формування корисних звичок користувача»

Як бачимо на діаграмі дерева вузлів, ми маємо п'ять основних бізнес-процесів. Кожен з цих процесів поділяється на бізнес-функції:

Реєстрація та автентифікація користувача

1. Перевірка валідності введених даних
2. Збереження нового користувача у базі
3. Видача токена доступу при успішній автентифікації

Створення та управління звичками

1. Створення нової звички
2. Редагування чи видалення звички
3. Збереження звички у базі даних

Відстеження виконання звичок

1. Фіксація факту виконання звички
2. Оновлення показників статистики
3. Передача даних для актуалізації прогресу

Надсилання нагадувань і сповіщень

1. Планування розкладу нагадувань

2. Надсилання push-сповіщення
3. Логування відправлених повідомлень

Актуалізація прогресу

1. Обчислення поточного рівня рослини
2. Зміна графічного стану у UI
3. Збереження стану у базі даних

4.3 Розробка моделі потоків даних системи

Відповідно до стандарту DFD розробимо модель потоків даних для того, щоб візуально представити всі процеси у системі, що пов'язані з даними.

Результат розробки наведено на рисунку 4.8.

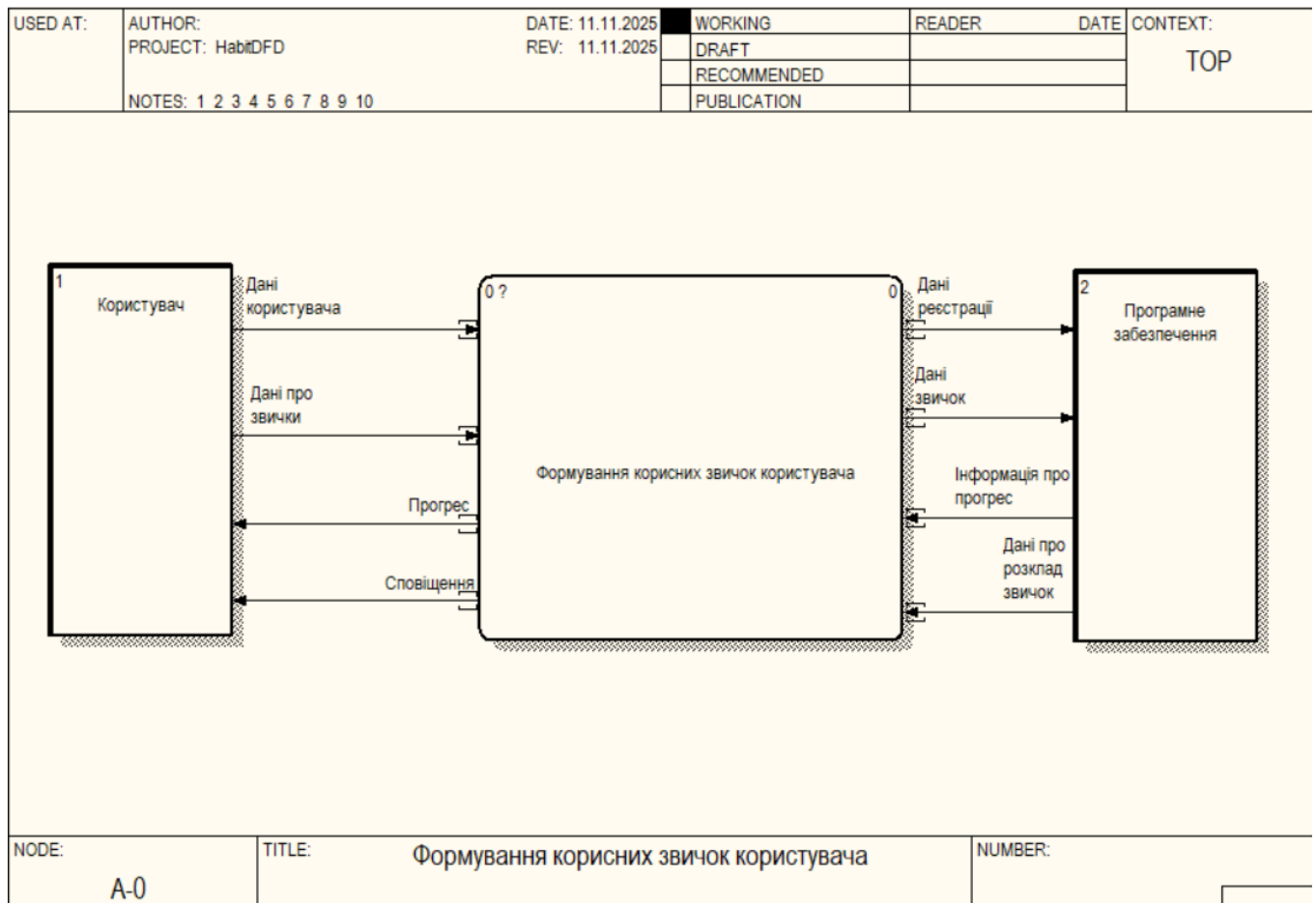


Рисунок 4.8 – Модель потоків даних системи

Після цього проведемо декомпозицію моделі потоків даних. Результат наведено на рисунку 4.9.

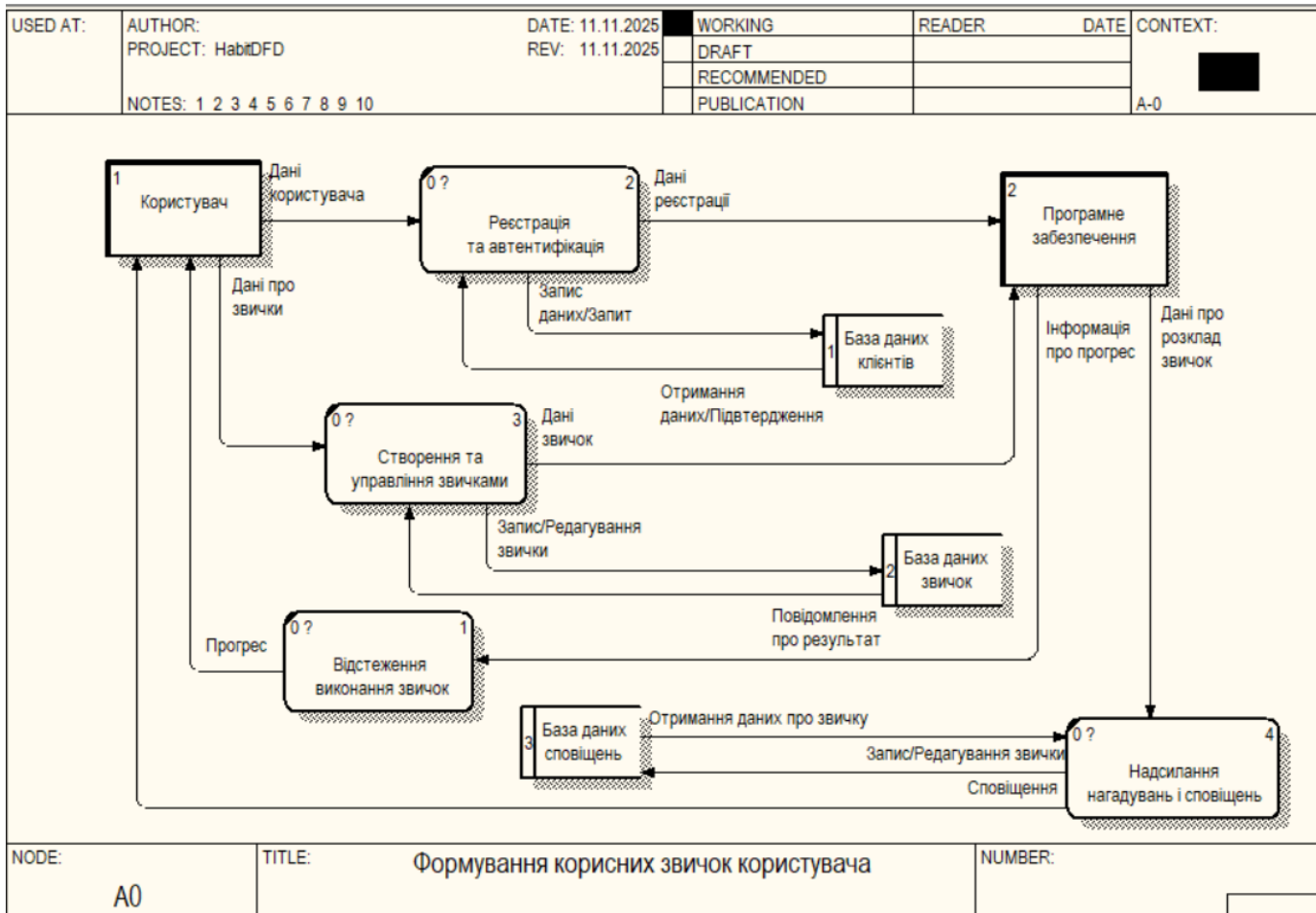


Рисунок 4.9 – Декомпозиція моделі потоків даних

4.4 Визначення функціональних вимог до клієнтської сторони мобільного застосунку

Клієнтська сторона мобільного застосунку HabitFlow, розробленою на платформі Android Studio, забезпечує основну взаємодію користувача з системою формування корисних звичок. У межах мобільного інтерфейсу реалізовано цілу низку функціональних модулів, кожен з яких відповідає за свій аспект користувацького досвіду — від перегляду щоденних завдань до аналізу статистики та кастомізації елементів гейміфікації. Доступ до окремих розділів застосунку залежить від статусу користувача: авторизований він чи ні, а також від того, які функції передбачені базовою або преміальною версією додатку.

Неавторизовані користувачі мають доступ лише до ознайомчого функціоналу — екрану онбордингу. На цьому екрані система вітає нового користувача, коротко презентує ідею та основні можливості HabitFlow, а також пропонує пройти авторизацію або реєстрацію. Це дозволяє забезпечити зручний перший контакт із застосунком та плавно підвести користувача до створення власного профілю. Після авторизації користувач отримує доступ до повного

набору функцій мобільного додатку. Основні екрани та їх можливості включають:

1. Сторінка звичок (головний екран роботи зі звичками)

Це один із ключових компонентів застосунку, який виконує роль центральної панелі користувацької діяльності.

На сторінці розташовано:

- Календар з відображенням поточної дати, що допомагає користувачу орієнтуватися в прогресі та зручніше планувати виконання звичок.
- Два основні розділи:
 - «Сьогодні» — відображає звички, заплановані саме на поточний день. Користувач може швидко відмітити виконання звички, що стимулює регулярність та створює ефект миттєвого досягнення.
 - «Всі» — надає можливість переглянути повний список створених звичок, незалежно від графіку виконання. Це корисно, якщо користувач хоче переглянути всі свої поточні практики або відредагувати одну з них.

2. Сторінка профілю

Ця сторінка є важливим елементом соціальної складової та персоналізації застосунку. На ній відображається:

- Фото профілю, ім'я та унікальний тег користувача.
- Блок потенційних друзів та активність друзів, зокрема їхні здобуті досягнення.
- Список друзів із можливістю швидкого переходу до їх профілів.
- Кнопки навігації для переходу до налаштувань та функції поділитися профілем, що сприяє соціальному поширенню застосунку.

3. Сторінка налаштувань

Цей екран забезпечує доступ до системних і персональних налаштувань:

- Базові параметри конфігурації (мовні налаштування, повідомлення, конфіденційність тощо).
- Можливість вийти з профілю.
- Окремий блок для переходу до оновлення акаунта до статусу Premium, що відкриває розширені можливості та додаткові функції.

4. Сторінка саду (елемент гейміфікації)

Сад — це візуальна метафора прогресу користувача. Кожна рослина уособлює окрему звичку. На цій сторінці відображаються всі рослини та коротка інформація щодо кожної з них.

Під час переходу до детального перегляду користувач може побачити:

- ім'я рослини, її поточний рівень та кількість досвіду;
- візуальний вигляд;
- поточний стрік виконання звички;
- найдовший стрік;
- кількість доступних предметів для кастомізації.

Також з цього екрану можна перейти до редагування рослини або відповідної звички.

5. Сторінка редагування рослини

Цей екран дає можливість користувачеві персоналізувати свій «рослинний аватар» звички.

На сторінці доступні операції:

- редагування імені рослини;
- вибір зовнішнього вигляду, зокрема предметів з категорій: капелюхи, горщики, аксесуари.

Це сприяє гейміфікації та посиленню мотивації за рахунок візуального прогресу.

6. Сторінка створення звички

Це функціональний модуль, що дозволяє користувачу створити нову звичку відповідно до власних цілей. Передбачені наступні параметри:

- назва та опис звички;
- вибір емоджі та кольору;
- частота виконання та тип звички;
- вибір часу дня (ранок, день, вечір або довільно);
- створення нагадування.

7. Сторінка редагування звички

За своїм виглядом даний екран повторює сторінку створення звички, але всі поля вже попередньо заповнені поточною інформацією. Це дозволяє легко та швидко вносити зміни в існуючу звичку.

8. Сторінка статистики

Це аналітичний модуль, який дає можливість користувачу оцінити свій прогрес. На сторінці відображається:

- календар виконання звичок;
- тижнева статистика активності;
- кращий стрік;
- звичка з найбільшим стріком;

- загальна кількість досягнень.

Функціонал цього екрану спрямований на підтримання довгострокової мотивації та усвідомлення власного прогресу.

Таким чином, клієнтська частина мобільного застосунку HabitFlow представляє собою комплексну систему, структуровану навколо потреб користувача у формуванні та підтриманні корисних звичок. Кожен екран виконує окрему роль у побудові комфортного, інтуїтивного та привабливого користувацького досвіду.

Наступним етапом проєкту стало створення діаграми класів, яка є одним із ключових інструментів моделювання в об'єктно-орієнтованому підході. Діаграма класів дає змогу формалізувати архітектуру мобільного застосунку, відобразивши основні сутності, їхні атрибути, методи та логічні зв'язки між ними. Такий підхід значно полегшує розуміння внутрішньої структури системи та забезпечує узгодженість на подальших етапах розробки.

У контексті створення мобільного додатку для формування корисних звичок із використанням елементів гейміфікації побудова діаграми класів дозволяє отримати чітке уявлення про те, як взаємодіють основні компоненти — від керування користувачами та їхніми звичками до відстеження прогресу, нагород та візуальних елементів гейміфікації. Візуалізація цих зв'язків сприяє більш структурованому та систематичному підходу до проєктування програмної логіки.

У результаті розробки була побудована діаграма класів, представлена на рисунку 4.10.

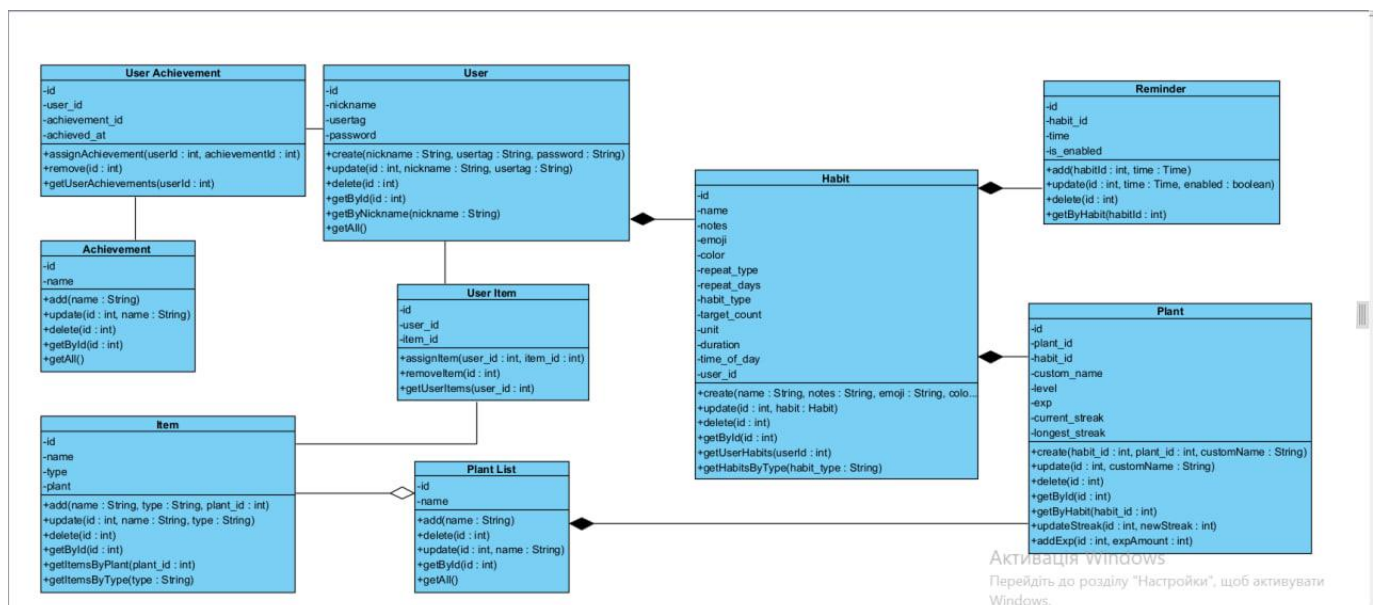


Рисунок 4.10 - Діаграма класів

Отримана діаграма відображає всі сутності, необхідні для функціонування мобільного застосунку, у вигляді окремих класів. Для кожного класу визначено основні атрибути, що містять дані про об'єкти, а також методи, які описують поведінку кожного елемента системи. Це надає цілісне та наочне уявлення про структуру застосунку, дозволяючи легко простежити, як інформація передається між класами та які залежності існують між ними.

Додатково на діаграмі відображено типи зв'язків між сутностями, що є важливими для розуміння ієрархії системи та ролі кожного компонента. Зокрема, позначено асоціації між класами, що демонструють сталі взаємовідносини (наприклад, зв'язок між користувачем та його звичками), агрегації, які вказують на структуровані об'єднання елементів (як у випадку з набором віртуальних нагород чи елементів гейміфікованих інтерфейсів), а також залежності — відображення взаємодій, що виникають лише в певних сценаріях.

Таким чином, створення діаграми класів стало важливим кроком у формуванні архітектури мобільного додатку. Вона не лише слугує інструментом документування, але й створює основу для логічної та технологічної реалізації системи, забезпечуючи цілісність, узгодженість і передбачуваність на всіх стадіях розробки.

5 ОПИС ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ ПРИ РОЗРОБЦІ СИСТЕМИ

5.1 Логічне і фізичне моделювання даних

Для розробки мобільного застосунку трекера звичок була використана мова програмування Java, яка зарекомендувала себе як надійний, стабільний та універсальний інструмент для створення додатків із розширеним функціоналом. Java забезпечує оптимальну продуктивність і підтримує велику кількість бібліотек, що значно спрощує розробку застосунків, орієнтованих на роботу з даними, статистикою, елементами гейміфікації та взаємодією з користувачем.

Як програмну платформу для зберігання й обробки інформації було обрано систему управління базами даних MySQL. Ця СУБД широко використовується в комерційних і наукових проєктах, що потребують надійного структурування та швидкого доступу до даних. У контексті роботи над трекером звичок MySQL дозволяє ефективно зберігати такі види інформації, як створені користувачами звички, дати їх виконання, динаміку прогресу, рівні мотивації, внутрішні нагороди та інші елементи гейміфікації, на основі яких формується користувацький досвід.

Вибір MySQL був обумовлений низкою суттєвих переваг:

- Відкритий код та безкоштовність. MySQL є відкритою та безкоштовною СУБД, що робить її доступною для використання в навчальних, дослідницьких та комерційних проєктах без необхідності придбання дорогих ліцензій. Такий підхід є особливо цінним під час розробки дипломного проєкту, оскільки дозволяє зосередитись на функціоналі застосунку, не витрачаючи ресурси на додаткове програмне забезпечення.
- Розвинена документація та підтримка спільноти. MySQL має велику базу офіційної документації та активну спільноту розробників. Це значно спрощує процес розробки, оскільки більшість технічних питань та можливих проблем уже мають готові рішення. При створенні трекера звичок це дозволяє швидко інтегрувати оптимальні формати зберігання статистики, календарних даних, щоденних позначок і механізмів відстеження прогресу.
- Швидкодія. Однією з ключових характеристик MySQL є її висока продуктивність. Для трекера звичок, де система часто звертається до бази даних для побудови графіків, аналізу статистики та визначення

нагород, швидка обробка запитів має велике значення. Користувач очікує миттєво отримувати інформацію про свій поточний прогрес, streak-и, виконані завдання чи зміни рівня мотивації.

- Масштабованість. У перспективі застосунок може розширюватися: зростатиме кількість користувачів, звичок, історичних даних та щоденних записів. MySQL добре масштабується, тому навіть при збільшенні кількості даних структура бази може ефективно обробляти більший потік запитів, не потребуючи значних змін в архітектурі застосунку.
- Підтримка стандартів SQL. Завдяки дотриманню міжнародних стандартів SQL MySQL забезпечує сумісність з іншими СУБД та інструментами. Це означає, що в майбутньому можлива міграція системи на інше програмне забезпечення або інтеграція з додатковими сервісами, якщо проєкт буде розширюватися або комерціалізуватися.
- Безпека. Оскільки трекер звичок працює з персональними даними користувачів — їхніми звичками, графіками дня, прогресом і приватними записами — важливо забезпечити належний рівень захисту. MySQL надає гнучкі механізми автентифікації та розмежування рівнів доступу, що запобігає несанкціонованому втручанню або втраті даних, а також підтримує засоби шифрування.
- Широка інтеграція. MySQL легко інтегрується з великою кількістю фреймворків, інструментів та бібліотек, що робить її універсальним вибором для складних або модульних проєктів. У випадку трекера звичок це дозволило без труднощів поєднати серверну частину з візуальними компонентами, логікою гейміфікації та підсистемою аналітики.

Після вибору СУБД наступним етапом стало побудова логічної моделі даних. Для цього була розроблена ER-діаграма відповідно до стандарту IDEF1X, який дозволяє формально та структуровано описати взаємозв'язки між основними сутностями системи. У контексті застосунку це такі сутності, як користувачі, звички, дні виконання, прогрес, нагороди, рівні, типи звичок та інші об'єкти, необхідні для повного функціонування системи відстеження й мотивації.

Створена ER-діаграма слугує основою для реалізації фізичної моделі бази даних, визначення ключів, обмежень цілісності та формування таблиць. Вона

забезпечує цілісне бачення структури системи та дозволяє уникнути логічних помилок у процесі реалізації. Результат на рисунку 5.1.

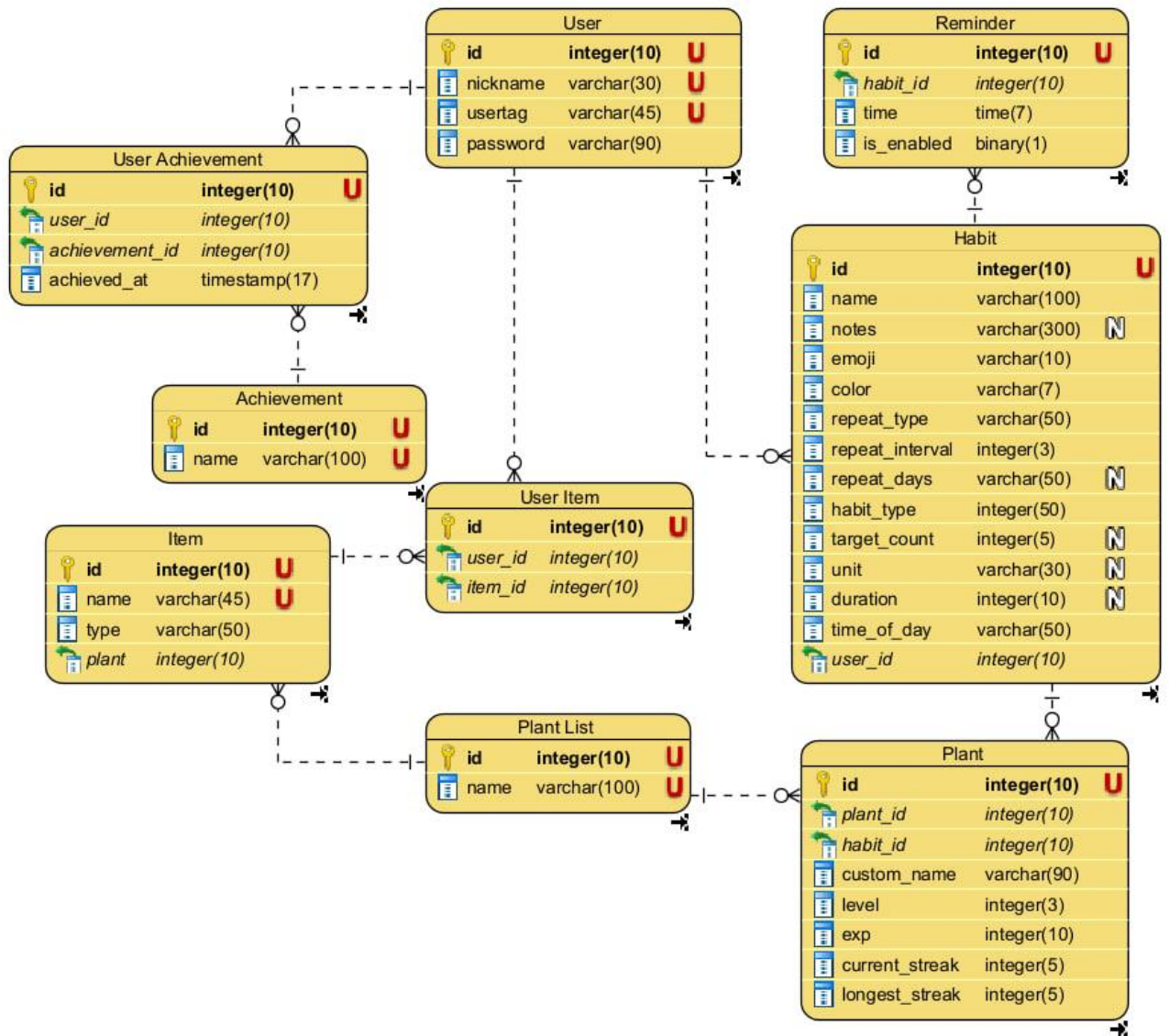


Рисунок 5.1 - Логічна модель даних відповідно до стандарту IDEF1X

Використовуючи логічну модель, побудуємо на її основі фізичну модель даних відповідно до стандарту IDEF1X. Результат на рисунку 5.2.

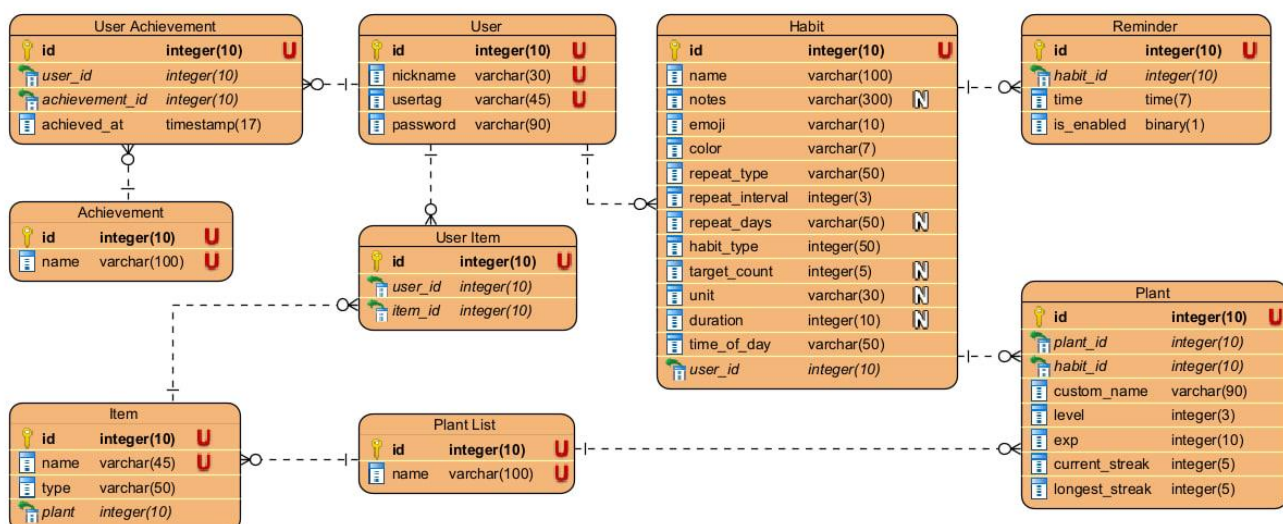


Рисунок 5.2 - Фізична модель даних відповідно до стандарту IDEF1X

У межах інформаційної системи HabitFlow використано реляційну модель даних, де кожна сутність представлена окремою таблицею з чітко визначеними атрибутами та зв'язками. Такий підхід забезпечує логічно структуроване зберігання інформації, мінімізує дублювання даних та спрощує їх подальшу обробку. Нижче наведено докладний опис кожної сутності бази даних.

Сутність achievement містить дані про всі можливі досягнення, які може отримати користувач у процесі використання застосунку. Кожне досягнення — це певний результат або нагорода за досягнення важливих етапів у виконанні звичок. Атрибути:

- id – первинний ключ. Цілочисельне поле типу INT із автоінкрементом. Є унікальним і ненульовим. Використовується для однозначної ідентифікації кожного досягнення в системі.
- name – назва досягнення. Атрибут типу VARCHAR(100). Містить текстову назву, наприклад: «7 днів поспіль», «Ранкова звичка прокачана», «Майстер таймера» тощо. Поле є унікальним, оскільки одна й та сама назва не повинна повторюватися серед різних досягнень.

Таблиця використовується як довідник для інших сутностей — зокрема, для user_achievement.

Сутність user містить базову інформацію про кожного користувача додатку HabitFlow. Ця таблиця є однією з ключових, бо через її первинний ключ пов'язуються майже всі інші сутності. Атрибути:

- `id` – первинний ключ. Тип `INT`, автоінкремент. Використовується як унікальний індекс користувача.
- `nickname` – нікнейм користувача. Тип `VARCHAR(30)`. Унікальний атрибут, що дозволяє користувачам мати власну унікальну назву профілю.
- `usertag` – тег користувача. Тип `VARCHAR(45)`. Це унікальний ідентифікатор, схожий на «користувацький ID» у Discord. Дозволяє відрізнити користувачів із однаковими нікнеймами.
- `password` – пароль користувача. Тип `VARCHAR(90)`. Тут зберігається хешований пароль для автентифікації.

Сутність `habit` є центральною в системі, адже саме вона описує всі звички, які створює користувач. Таблиця містить розширений набір атрибутів, що дозволяє точно налаштувати параметри звички, включно з повтореннями, типом та візуальним оформленням. Атрибути:

- `id` – первинний ключ. Тип `INT`, автоінкремент.
- `name` – назва звички. Тип `VARCHAR(100)`. Наприклад: «Випити воду», «Пробіжка», «Саморозвиток».
- `notes` – додаткові нотатки. Тип `VARCHAR(300)`, може бути порожнім. Використовується для збереження описів чи особистих уточнень.
- `emojı` – емодзі, пов'язаний зі звичкою. Тип `VARCHAR(10)`. Використовується для візуальної ідентифікації.
- `color` – колір відображення звички. Тип `VARCHAR(7)`. Зазвичай містить HEX-код, наприклад: `#1D48EA`.
- `repeat_type` – тип повторення. Тип `ENUM('daily','weekly','monthly')`. Дає змогу задавати логіку виконання.
- `repeat_interval` – інтервал повторення. Тип `INT`. Наприклад, повторення раз на 2 дні чи раз на 3 тижні.
- `repeat_days` – конкретні дні повторення. Тип `VARCHAR(50)`. Використовується для `weekly`-звичок(`MON`, `TUE`) або `monthly`-звичок(1, 5, 12).
- `habit_type` – тип звички. `ENUM('counter','timer')`.

Приклад:

– `counter`: «випити 5 склянок води»

– `timer`: «читати 20 хвилин»

- `target_count` – ціль за кількістю (для `counter`). Тип `INT`, може бути `NULL`.

- `unit` – одиниця вимірювання. Наприклад: «разів», «кроків», «хвилин».
- `duration` – тривалість (для `timer`). Тип `INT`.
- `time_of_day` – рекомендований час.
ENUM('morning','afternoon','evening','anytime').
- `user_id` – зовнішній ключ на таблицю `user`. Тип `INT`. Пов'язує звичку з її власником.

Таблиця має один із найбільших наборів параметрів у системі `HabitFlow`, оскільки кожна звичка має різну поведінку, і ці атрибути регулюють її логіку.

Сутність `plant_list` служить для збереження інформації про всі доступні види рослин, які можуть бути пов'язані зі звичками користувачів і рости відповідно до прогресу. Атрибути:

- `id` – первинний ключ, `INT`, автоінкремент.
- `name` – назва рослини, `VARCHAR(100)`.
Наприклад: «Алое», «Сукулент», «Бамбук».

Це довідкова таблиця, яка використовується у `plant` та `item`.

Сутність `item` містить дані про косметичні елементи (аксесуар, шляпа, горщик), які користувач може надягати на рослини.

Атрибути:

- `id` – первинний ключ.
- `name` – назва предмета, `VARCHAR(45)`. Унікальна, наприклад: «Cowboy Hat», «Golden Pot».
- `type` – категорія предмета. ENUM('hat','pot','accessory').
- `plant` – зовнішній ключ на `plant_list`. Указує, для якої рослини цей предмет доступний.

Таблиця `plant` зберігає дані про конкретну рослину, прив'язану до конкретної звички та користувача. Кожен користувач може мати багато рослин, залежно від кількості звичок. Атрибути:

- `id` – первинний ключ.
- `plant_id` – зовнішній ключ на таблицю `plant_list`.
- `habit_id` – зовнішній ключ на `habit`. Вказує, за виконання якої звички рослина «росте».
- `custom_name` – назва рослини, яку задав користувач.
- `level` – рівень рослини. Тип `INT`, початкове значення: 1.

- `exp` – досвід рослини. Зростає щоразу, коли користувач виконує звичку.
 - `current_streak` – поточний стрик. Відображає кількість днів поспіль.
 - `longest_streak` – найдовший стрик за всю історію.
- Сутність є однією з ключових елементів гейміфікації.

Таблиця `reminder` служить для зберігання нагадувань до звичок. Атрибути:

- `id` – первинний ключ.
- `habit_id` – зовнішній ключ на `habit`. Нагадування завжди належить певній звичці.
- `time` – час нагадування (TIME).
- `is_enabled` – активність нагадування TINYINT(1), значення 0 або 1.

Таблиця `user_achievement` - це зв'язувальна таблиця «багато-до-багатьох» між `user` і `achievement`. Атрибути:

- `id` – первинний ключ.
- `user_id` – зовнішній ключ на `user`.
- `achievement_id` – зовнішній ключ на `achievement`.
- `achieved_at` – дата та час отримання досягнення. Автоматично встановлюється поточною датою.

Таким чином можна легко відстежувати історію прогресу користувача.

Таблиця `user_item` зберігає предмети, які користувач отримав або розблокував. Атрибути:

- `id` – первинний ключ.
- `user_id` – зовнішній ключ на `user`.
- `item_id` – зовнішній ключ на `item`.

Ця таблиця відповідає за систему кастомізації рослин.

5.2 Створення бази даних на платформі СУБД

Наступним важливим етапом розробки компонентів інформаційної системи для застосунку трекара звичок стало проектування та створення структурованої бази даних на платформі СУБД MySQL. Оскільки система має працювати зі значною кількістю даних користувачів — зокрема інформацією про створені звички, їхню періодичність, прогрес, статистичні показники, рівень залучення та елементи гейміфікації — виникла потреба забезпечити зручне,

надійне та гнучке керування цими даними. Саме тому було обрано MySQL, як одну з найбільш стабільних, поширених та оптимізованих систем управління базами даних, що підтримує масштабованість і працює ефективно навіть під значним навантаженням.

Для роботи з базою даних було використано програмне середовище MySQL Workbench — інструмент, який надає широкі можливості для візуального моделювання, документування та подальшого адміністрування баз даних. Завдяки інтуїтивному інтерфейсу та підтримці ER-діаграм, Workbench дозволив детально спроектувати логічну структуру майбутньої бази даних, враховуючи особливості функціоналу трекера звичок: від зберігання створених користувачами звичок та історії їх виконання до фіксації нагород, рівнів і внутрішніх механік гейміфікації.

Особливо корисною під час розробки стала можливість зворотної розробки. Після створення таблиць та встановлення зв'язків між ними система автоматично сформувала детальну ER-діаграму, яка наочно відображає всю структуру бази даних та взаємозв'язки між ключовими сутностями. Такий підхід не лише спростив документування бази даних для дипломного проєкту, а й дозволив переконатися у правильності побудови логічних зв'язків, відповідності структури вимогам функціоналу та можливостям подальшого масштабування системи. Отриману діаграму наведено на рисунку 5.3.

Після реалізації фізичної моделі бази даних наступним кроком стало визначення посилальної цілісності, що є критично важливим аспектом для коректної роботи інформаційної системи. Було проаналізовано всі зовнішні ключі, встановлені між таблицями, обрано оптимальні режими каскадного оновлення та видалення, а також забезпечено узгодженість даних під час виконання операцій CRUD. Це дозволяє гарантувати, що дані про звички, виконання, щоденні позначки, нагороди чи рівні користувача не втратять зв'язків і не стануть несумісними у випадках оновлення або видалення пов'язаних записів. Повний список установлених обмежень посилальної цілісності наведено у таблиці 5.1.

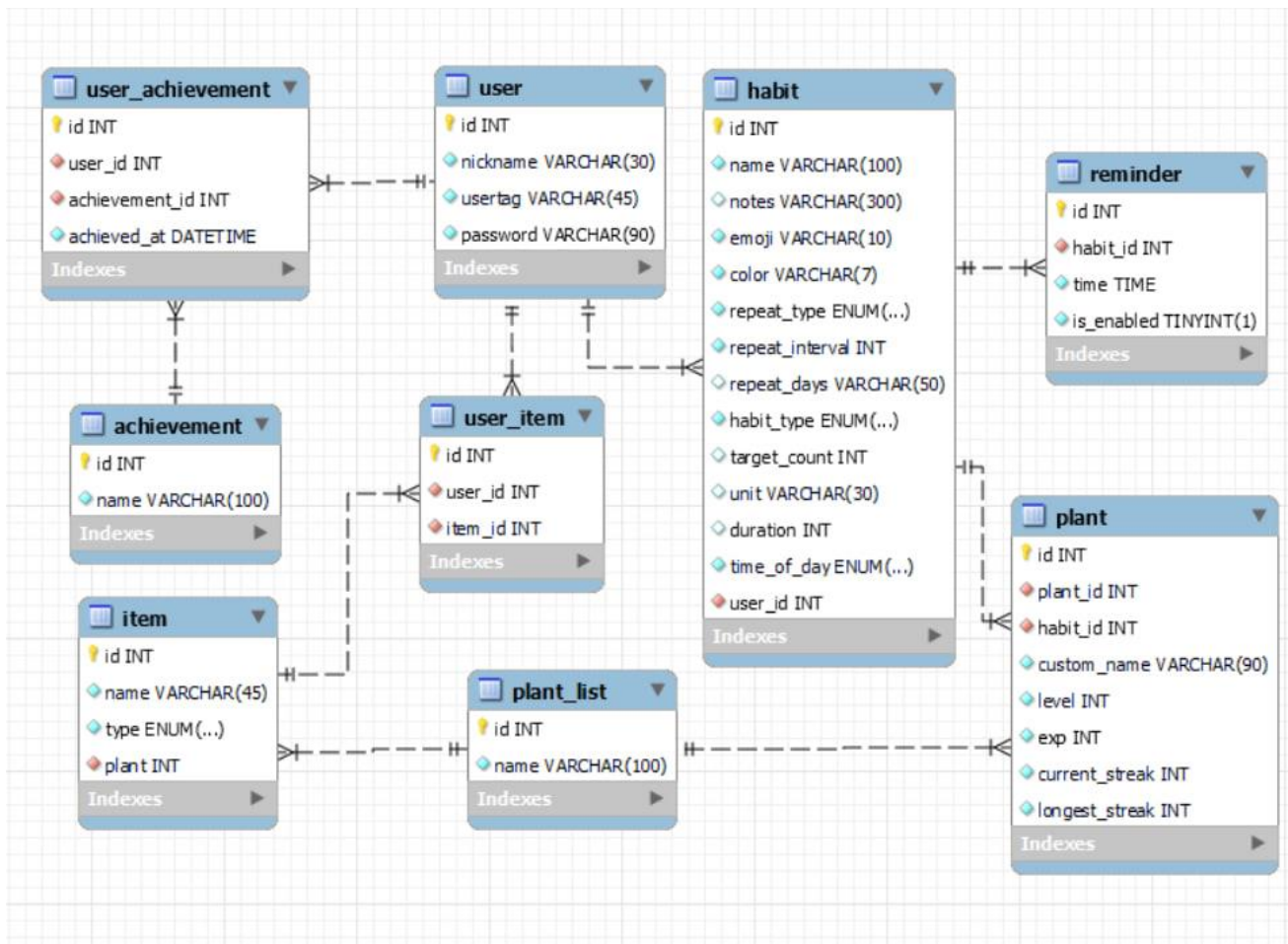


Рисунок 5.3 – Модель даних

Таблиця 5.1 - Список обмежень посилальної цілісності

| № | Таб. 1, зовнішній ключ | Таб. 2, первинний ключ | Посилальна цілісність для оновлення | Посилальна цілісність для видалення |
|---|----------------------------------|------------------------|-------------------------------------|-------------------------------------|
| 1 | habit, user_id | user, id | CASCADE | CASCADE |
| 2 | item, plant | plant_list, id | CASCADE | CASCADE |
| 3 | plant, plant_id | plant_list, id | CASCADE | CASCADE |
| 4 | plant, habit_id | habit, id | CASCADE | CASCADE |
| 5 | reminder, habit_id | habit, id | CASCADE | CASCADE |
| 6 | user_achievement, achievement_id | achievement, id | CASCADE | CASCADE |
| 7 | user_achievement, user_id | user, id | CASCADE | CASCADE |
| 8 | user_item, user_id | user, id | CASCADE | CASCADE |

| | | | | |
|---|--------------------|----------|---------|---------|
| 9 | user_item, item_id | item, id | CASCADE | CASCADE |
|---|--------------------|----------|---------|---------|

5.3 Розробка алгоритму роботи системи

Для забезпечення цілісного розуміння внутрішньої логіки функціонування мобільного застосунку HabitFlow було прийнято рішення розробити мапу застосунку (App Map). На відміну від класичного sitemap, що зазвичай використовується у веб-проектах, App Map є більш доречним підходом саме для мобільних систем, оскільки він дозволяє відобразити повну структуру екранів, логіку переходів та взаємозв'язки між основними компонентами інтерфейсу.

Мапа застосунку — це структурована схема, яка включає всі екрани мобільного додатка та показує, яким чином користувач може переміщуватися між ними у процесі взаємодії. Вона виконує роль візуальної моделі роботи програми, що дозволяє зрозуміти логіку переходів, визначити ключові маршрути користувача та виокремити функціональні блоки системи. Таке графічне представлення є надзвичайно корисним на етапі проектування, оскільки допомагає впорядкувати структуру інтерфейсу, уникнути дублювання або надмірної складності навігації, а також забезпечити послідовність і зручність користувацького досвіду.

Зазвичай мапа мобільного застосунку будується за деревоподібним принципом, де головний екран виступає кореневим елементом, а всі інші екрани та підекрани формують відповідні гілки навігації. Для HabitFlow таким коренем є головна домашня сторінка додатка, з якої користувач може перейти до розділів перегляду звичок, створення нових звичок, перегляду прогресу, налаштувань профілю, системи гейміфікації тощо.

Створення App Map стало важливим етапом у розробці даного мобільного застосунку, оскільки воно дозволило сформуванню чіткого уявлення про структуру проекту ще до етапу безпосереднього програмування. Це значно спростило процес імплементації інтерфейсу, полегшило організацію навігації всередині Android-дodatка та стало основою для подальшого проектування бізнес-логіки.

Мапу застосунку HabitFlow, яка була розроблена в рамках даної роботи, наведено на рисунку 5.4.

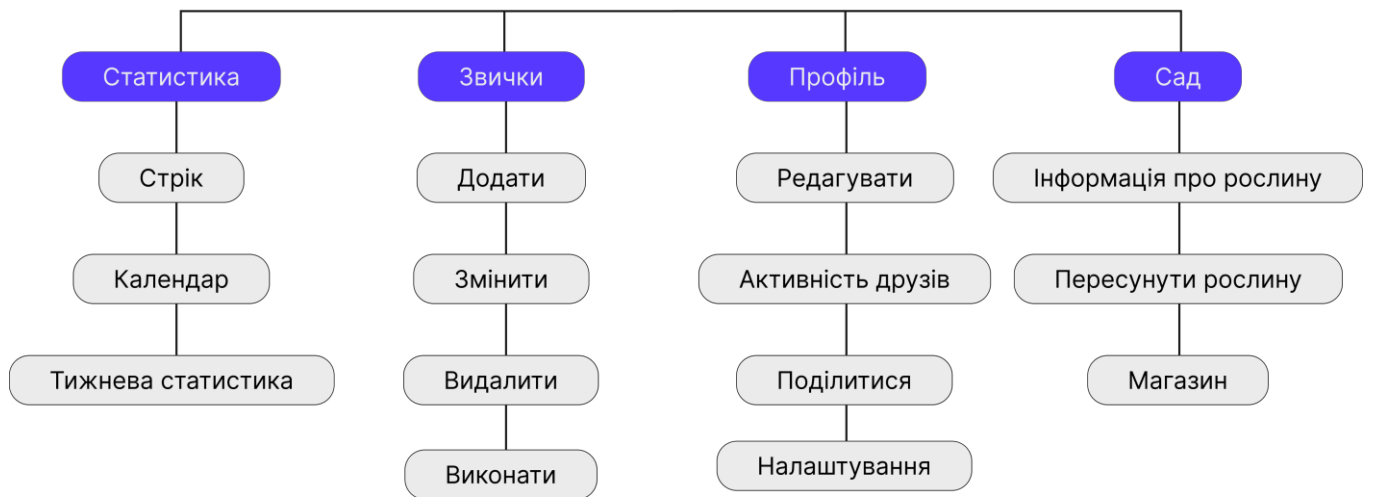


Рисунок 5.4 – Мапа додатку

5.4 Тестування розробленого програмного забезпечення

Останнім етапом життєвого циклу розробки інформаційної системи є її тестування.

Цей етап має ключове значення, оскільки дозволяє оцінити якість програмного продукту, переконатися у правильності реалізації бізнес-логіки та виявити помилки, які могли бути пропущені на попередніх стадіях. Тестування підтверджує, що система поводить відповідно до вимог, задекларованих на етапах аналізу та проектування, та гарантує стабільність роботи додатку при подальшій експлуатації кінцевими користувачами.

Для перевірки коректності реалізованого функціоналу було створено окремий модуль із тестовими класами, які відповідають за тестування найважливіших методів доступу до даних. Оскільки програмна система використовує шаблон доступу до бази даних DAO (Data Access Object), основна логіка взаємодії з даними зосереджена саме в DAO-класах. Тому тестування охоплює такі операції: додавання нового запису, оновлення існуючих даних, видалення та отримання інформації з бази.

Для виконання інтеграційних тестів було використано ізольовану тестову базу даних, яка повністю повторює структуру основної бази, але не містить реальних користувацьких даних. Такий підхід дозволяє проводити тестування без ризику пошкодження основної інформації, а також гарантує відтворюваність результатів. Перед виконанням кожного тесту здійснюється ініціалізація

тестового середовища: підключення до тестової БД, створення об'єктів DAO та очищення таблиць.

Нижче наведено приклад одного з тестових класів, який був розроблений для перевірки функціональності класу HabitDAO. Він містить тести для операцій додавання, читання, оновлення та видалення записів. Кожен тест виконує реальну SQL-операцію над тестовою базою даних, а потім перевіряє коректність змін за допомогою SELECT-запитів.

Тестування дозволяє переконатися у правильній роботі таких методів:

- `addHabit()` — перевіряє можливість успішного додавання нового запису в таблицю `habit`. Після виконання методу тест здійснює запит до бази та перевіряє, що нова звичка дійсно була збережена.
- `getHabitsByUserId()` — тестує здатність системи повертати всі звички, які належать певному користувачу. Метод має коректно формувати список об'єктів на основі даних з БД.
- `updateHabit()` — перевіряє, чи коректно оновлюються значення існуючого запису. Після оновлення тест повторно зчитує дані та порівнює їх із очікуваними.
- `deleteHabitById()` — перевіряє, чи успішно видалається запис із бази даних. Тест підтверджує, що після виконання методу відповідний запис більше не існує у таблиці `habit`.

Лістинг 5.1 - Тестування функціоналу класу звичок

```
package org.nure.dao;
import org.junit.jupiter.api.*;
import org.nure.DatabaseTestConnection;
import org.nure.entity.Habit;
import org.nure.enums.*;

import java.sql.Connection;
import java.util.List;

import static org.junit.jupiter.api.Assertions.*;

@TestMethodOrder(MethodOrderer.OrderAnnotation.class)
class HabitDAOTest {

    private static Connection connection;
    private static HabitDAO habitDAO;

    @BeforeAll
    static void setup() throws Exception {
```

```

        connection = DatabaseTestConnection.getConnection();
        habitDAO = new HabitDAO(connection);
    }

    @Test
    @Order(1)
    void testAddHabit() throws Exception {
        Habit habit = new Habit(
            0,
            "Test Habit",
            "Notes",
            "🔥",
            "#FF0000",
            repeat_type.daily,
            1,
            "Mon,Tue",
            habit_type.counter,
            1,
            "times",
            0,
            time_of_day.morning,
            1
        );

        assertDoesNotThrow(() -> habitDAO.addHabit(habit));
    }

    @Test
    @Order(2)
    void testGetHabitsByUser() throws Exception {
        List<Habit> habits = habitDAO.getHabitsByUserId(1);

        assertNotNull(habits);
        assertTrue(habits.size() > 0);
    }

    @Test
    @Order(3)
    void testUpdateHabit() throws Exception {
        List<Habit> habits = habitDAO.getHabitsByUserId(1);
        Habit habit = habits.get(0);

        habit.setName("Updated Habit");

        assertDoesNotThrow(() -> habitDAO.updateHabit(habit));
    }

    @Test
    @Order(4)

```

```

void testDeleteHabit() throws Exception {
    List<Habit> habits = habitDAO.getHabitsByUserId(1);
    Habit habit = habits.get(0);

    assertDoesNotThrow(() ->
habitDAO.deleteHabitById(habit.getId()));
    }
}

```

Результат тестування наведено на рисунку 5.5.

| Test Method | Execution Time |
|-----------------------|----------------|
| testAddHabit() | 94 ms |
| testGetHabitsByUser() | 86 ms |
| testUpdateHabit() | 20 ms |
| testDeleteHabit() | 12 ms |

Рисунок 5.5 – Результат тестування

Кожен тест був реалізований у схожому стилі, з чіткою структурою та однозначним критерієм успішності. Завдяки таким тестам вдалося підтвердити правильність роботи CRUD-операцій (Create, Read, Update, Delete), що є фундаментальною частиною логіки системи HabitFlow.

Під час тестування також було виявлено декілька структурних неточностей, пов'язаних із відповідністю назв таблиць у SQL-запитах та фактичних назв таблиць у базі даних. Після коригування структури тестової БД та синхронізації імен проблеми було усунуто. Це ще раз демонструє важливість тестування для виявлення прихованих помилок, які не очевидні під час написання основного коду.

Таким чином, проведення тестування дозволило підтвердити стабільність роботи підсистеми доступу до даних та її відповідність очікуваним вимогам. Розроблені тести можуть і надалі використовуватися під час розширення функціоналу для запобігання регресійним помилкам.

5.5 Розробка тригерів серверної частини системи

Одним із завершальних етапів розробки інформаційної системи HabitFlow стало написання тригерів MySQL, які забезпечують додатковий рівень контролю та цілісності даних. Незважаючи на те, що серверна частина системи реалізує більшість перевірок у Java-кодi, тригери відіграють важливу роль як незалежний механізм захисту від некоректних операцій.

Структура бази даних містить низку таблиць, пов'язаних між собою зовнішніми ключами, але у деяких випадках необхідно посилити логіку валідності даних саме на рівні СУБД. Особливо це стосується:

- створення звичок (habit),
- прив'язки нагадувань (reminder),
- запису прогресу рослин (plant),
- видачі досягнень користувачам (user_achievement),
- надання користувачеві предметів (user_item).

Нижче розглянуто основні тригери, розроблені для системи.

Перевірка існування користувача при створенні звички

Таблиця habit містить поле user_id, що вказує на власника звички. Хоча це поле захищене зовнішнім ключем, тригер дозволяє перехопити помилку до виконання SQL-операції та повернути більш зрозуміле повідомлення.

Лістинг 5.2 – Тригер для перевірки існування користувача

```
CREATE TRIGGER habit_BEFORE_INSERT
BEFORE INSERT ON habit
FOR EACH ROW
BEGIN
    DECLARE user_exists INT;

    SELECT COUNT(*) INTO user_exists
    FROM user
    WHERE id = NEW.user_id;

    IF user_exists = 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Cannot create habit: user does not
exist';
```

```
END IF;  
END;
```

Перевірка коректності створення нагадування

Кожне нагадування прив'язується до певної звички через habit_id.

Лістинг 5.3 – Тригер для перевірки коректності створення нагадування

```
CREATE TRIGGER reminder_BEFORE_INSERT  
BEFORE INSERT ON reminder  
FOR EACH ROW  
BEGIN  
    DECLARE habit_exists INT;  
  
    SELECT COUNT(*) INTO habit_exists  
    FROM habit  
    WHERE id = NEW.habit_id;  
  
    IF habit_exists = 0 THEN  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT = 'Cannot create reminder: habit does  
not exist';  
    END IF;  
END;
```

Валідація прогресу рослини

У таблиці plant кожен запис містить:

- plant_id — посилання на plant_list,
- habit_id — звичку, що "вирощує" рослину.

Тригер перевіряє обидва зв'язки.

Лістинг 5.4 – Тригер для валідації прогресу рослини

```
CREATE TRIGGER plant_BEFORE_INSERT  
BEFORE INSERT ON plant  
FOR EACH ROW  
BEGIN  
    DECLARE plant_type_exists INT;  
    DECLARE habit_exists INT;  
  
    SELECT COUNT(*) INTO plant_type_exists
```

```

FROM plant_list
WHERE id = NEW.plant_id;

IF plant_type_exists = 0 THEN
    SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Invalid plant type';
END IF;

SELECT COUNT(*) INTO habit_exists
FROM habit
WHERE id = NEW.habit_id;

IF habit_exists = 0 THEN
    SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Invalid habit selected for plant';
END IF;
END;

```

Видача досягнень користувачам

Таблиця user_achievement містить два ключові поля:

- user_id
- achievement_id

Тригер контролює існування обох сутностей.

Лістинг 5.5 – Тригер для контролю видачі досягнень

```

CREATE TRIGGER user_achievement_BEFORE_INSERT
BEFORE INSERT ON user_achievement
FOR EACH ROW
BEGIN
    DECLARE user_exists INT;
    DECLARE achievement_exists INT;

    SELECT COUNT(*) INTO user_exists
    FROM user
    WHERE id = NEW.user_id;

    IF user_exists = 0 THEN
        SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Invalid user';
    END IF;

```

```

SELECT COUNT(*) INTO achievement_exists
FROM achievement
WHERE id = NEW.achievement_id;

IF achievement_exists = 0 THEN
    SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Invalid achievement';
END IF;
END;

```

Перевірка валідності предметів користувача

Лістинг 5.6 – Тригер для перевірки валідності предметів

```

CREATE TRIGGER user_item_BEFORE_INSERT
BEFORE INSERT ON user_item
FOR EACH ROW
BEGIN
    DECLARE user_exists INT;
    DECLARE item_exists INT;

    SELECT COUNT(*) INTO user_exists
    FROM user
    WHERE id = NEW.user_id;

    IF user_exists = 0 THEN
        SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Invalid user';
    END IF;

    SELECT COUNT(*) INTO item_exists
    FROM item
    WHERE id = NEW.item_id;

    IF item_exists = 0 THEN
        SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Invalid item';
    END IF;
END;

```

ВИСНОВКИ

У процесі виконання роботи було досліджено, спроектовано та частково реалізовано інформаційну систему для формування та підтримки корисних звичок із використанням механізмів гейміфікації. Система поєднує функціональність трекера звичок із візуальними ігровими елементами, що стимулюють регулярне виконання дій — зокрема, концепцію «вирощування рослин», стан яких залежить від активності користувача.

У ході дослідження проаналізовано наявні аналоги, їхні сильні та слабкі сторони, а також сучасні тенденції у сфері гейміфікації. Виявлено, що значна частина існуючих рішень зосереджена лише на статистиці та нагадуваннях, проте не завжди забезпечує емоційне залучення користувачів. Запропонована система вирішує цю проблему шляхом поєднання прогресу звички з візуальним зростанням віртуальної рослини, що створює додатковий рівень мотивації та формує відчуття відповідальності за результат.

На основі проведеного аналізу було сформовано модель бізнес-процесів, включно з плануванням нагадувань, оновленням статистики, зміною стану рослини та іншими ключовими етапами взаємодії з користувачем. Для кожного процесу визначено обмеження, ресурси та контроль, а також описано системні й законодавчі фактори, що впливають на їх виконання. Це забезпечило формування цілісної архітектури майбутнього додатку та дозволило розробити логіку як користувацької, так і внутрішньої взаємодії компонентів.

Розроблена концепція демонструє ефективність застосування методів програмної інженерії, CASE-технологій, UML-моделювання та структурного аналізу під час побудови систем, орієнтованих на зміну поведінки. Використання гейміфікації у вигляді “росту рослини” показує високу потенційну результативність у підвищенні регулярності виконання звичок, покращенні користувацького досвіду та створенні позитивної емоційної взаємодії із застосунком.

Таким чином, розроблена система може слугувати основою для повноцінного мобільного чи веб-додатку, здатного допомогти людям формувати корисні звички через поєднання інформативності, зручності та ігрової мотивації. Надалі доцільним є розширення можливостей профілю, впровадження системи очок, рівнів, соціальних елементів та персоналізованих рекомендацій — це дозволить ще більше підвищити залученість та ефективність роботи додатку.

ПЕРЕЛІК ДЖЕРЕЛ

1. Фог Б. Формула Фога: Управління поведінкою за допомогою малих змін. Київ : Наш Формат, 2020. 224 с.
2. Werbach K., Hunter D. For the Win: How Game Thinking Can Revolutionize Your Business. Wharton Digital Press, 2012. — 148 p.
3. Hamari J., Koivisto J., Sarsa H. Does Gamification Work? – A Literature Review of Empirical Studies. Proceedings of the 47th Hawaii International Conference on System Sciences (HICSS), 2014, pp. 3025–3034.
4. Sailer M., Hense J., Mayr S., Mandl H. How Gamification Motivates: An Experimental Study of the Effects of Specific Game Design Elements on Psychological Need Satisfaction. Computers in Human Behavior, vol. 69, 2017, pp. 371–380.
5. Johnson D., Deterding S., Kuhn K., Staneva A., Stoyanov S., Hides L. Gamification for Health and Wellbeing: A Systematic Review. Internet Interventions, vol. 6, 2016, pp. 89–106.
6. Edwards E.A., Lumsden J., Rivas C., Steed L., Edwards L.A., Thiyagarajan A., et al. Gamification for Health Promotion: Systematic Review of Behaviour Change Techniques in Smartphone Apps. BMJ Open, vol. 6, no. 10, 2016.
7. Cugelman B. Gamification: What It Is and Why It Matters to Digital Health Behavior Change Developers. JMIR Serious Games, vol. 1, no. 1, 2013.
8. Orji R., Moffatt K. Persuasive Technology for Health and Wellness: State-of-the-Art and Future Directions. Journal of Medical Internet Research, vol. 20, no. 4, 2018.
9. Arias J., Bălău A., Tăucean I. Gamification in Enterprise Systems: A Conceptual Framework for Integrating Game Elements into Enterprise Architecture. Procedia Computer Science, vol. 176, 2020, pp. 3049–3058.
10. Jain P., Kumar Y., Singh R. Machine Learning for Mobile Health Applications: A Survey. Mobile Information Systems, 2021, Article ID 9990163.
11. Deterding S., Dixon D., Khaled R., Nacke L. From Game Design Elements to Gamefulness: Defining "Gamification". Proceedings of the 15th International Academic MindTrek Conference, 2011, pp. 9–15.
12. ДЕТЕРДИНГ С. Gamification: Designing for Intrinsic Motivation. Interaction Design Review. 2018. No 2. С. 15–27.

13. Hamari J., Koivisto J., Sarsa H. Does Gamification Work? – A Literature Review of Empirical Studies. Proceedings of the 47th Hawaii International Conference on System Sciences. 2014. P. 3025–3034.
14. Seaborn K., Fels D. Gamification in Theory and Action: A Survey. International Journal of Human-Computer Studies. 2015. Vol. 74. P. 14–31.
15. Landers R. N. Developing a Theory of Gamified Learning: Linking Serious Games and Gamification of Learning. Simulation & Gaming. 2014. Vol. 45, No 6. P. 752–768.
16. Wood W., Neal D. A New Look at Habits and the Habit–Goal Interface. Psychological Review. 2016. Vol. 123, No 4. P. 325–340.
17. Gardner B. A Review and Analysis of the Use of “Habit” in Health Behaviour Research. Psychology & Health. 2015. Vol. 30, No 3. P. 336–368.
18. Chou Y.-K. Actionable Gamification: The Octalysis Framework. Fremont : Octalysis Media, 2015. 320 p.
19. Deci E., Ryan R. Intrinsic and Extrinsic Motivation. Contemporary Educational Psychology. 2000. Vol. 25. P. 54–67.
20. Melton A., Ruiz P. Gamification Structures in Mobile Habit-Tracking Applications. Proceedings of the 11th IEEE Int. Conf. on Human-Centric Computing. Tokyo, 2020. P. 122–126.
21. Kowal M. Gamification for Behavioral Change: Trends and Challenges. International Journal of Digital Psychology. 2022. No 1. P. 62–74.
22. Буряк О., Савчук Н. UX-дизайн та гейміфікація: інструменти залучення користувачів. Київ : КНЕУ, 2020. 148 с.
23. Хоменко Л. Гейміфікація як інструмент підвищення залученості користувачів у мобільних додатках. Вісник КНУ «Інформатика та системи». 2021. No 3. С. 48–56.
24. Gamification in Behavior Management: Research Review. London : Wiley Research Group, 2019. 110 p.
25. Habit Tracking Apps Market Report 2023. San Francisco : Digital Trends Analytics, 2023. 87 p.
26. Clear J. Atomic Habits: An Easy & Proven Way to Build Good Habits & Break Bad Ones. New York: Avery, 2018. 320 p.
27. Duhigg Ch. The Power of Habit: Why We Do What We Do in Life and Business. New York: Random House, 2012. 371 p.
28. Eyal N., Hoover R. Hooked: How to Build Habit-Forming Products. New York: Portfolio/Penguin, 2014. 256 p.

29. Ryan R. M., Deci E. L. *Self-Determination Theory: Basic Psychological Needs in Motivation, Development, and Wellness*. New York: Guilford Press, 2017. 756 p.
30. Michie S., van Stralen M. M., West R. *The Behaviour Change Wheel: A New Method for Characterising and Designing Behaviour Change Interventions*. *Implementation Science*. 2011. Vol. 6, Article 42.
31. Fogg B. J. *A Behavior Model for Persuasive Design*. *Proceedings of the 4th International Conference on Persuasive Technology*. 2009. P. 1–7.
32. Hollis V., Konrad A., Whittaker S. *Change of Heart: Emotion Tracking to Promote Behavior Change*. *CHI '17: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2017. P. 3383–3395.
33. Looyestyn J., Kernot J., Boshoff K., Ryan J., Edney S., Maher C. *Does Gamification Increase Engagement with Online Programs? A Systematic Review*. *PLOS ONE*. 2017. Vol. 12(3): e0173403.
34. Koivisto J., Hamari J. *The Rise of Motivational Information Systems: A Review of Gamification Research*. *International Journal of Information Management*. 2019. Vol. 45. P. 191–210.
35. Chung P. K., Lin S. H. *Habit Formation in Mobile Health Applications: A Systematic Review*. *Journal of Medical Internet Research (JMIR)*. 2021. Vol. 23(4): e23790.