ЗМІСТ

[1 РОЛІ КОРИСТУВАЧІВ В ІНФОРМАЦІЙНІЙ СИСТЕМІ 2](#_Toc470450044)

[2 USE CASE UML ДІАГРАМИ 3](#_Toc470450045)

[3 ДІАГРАМИ ПОСЛІДОВНОСТЕЙ 4](#_Toc470450046)

[4 IDEF3 5](#_Toc470450047)

[5 МОДЕЛІ ЖИТТЄВИХ ЦИКЛІВ РОЗРОБКИ ОСНОВНИХ ПРОЦЕСІВ СИСТЕМИ 6](#_Toc470450048)

[6 DFD 7](#_Toc470450049)

[7 ERD 8](#_Toc470450050)

[8 АРХІТЕКТУРИ СИСТЕМИ ДЛЯ ОСНОВНИХ ПРОЦЕСІВ ІС 10](#_Toc470450051)

# 1 РОЛІ КОРИСТУВАЧІВ В ІНФОРМАЦІЙНІЙ СИСТЕМІ

У ході аналізу інформаційної системи, було визначено наступних користувачів:

а) неавторизований користувач;

б) авторизований користувач(студент);

в) авторизований користувач(викладач);

г) адміністратор баз даних.

Неавторизований користувач має наступні можливості: реєстрація.

Студенти у якості області видимості даних, мають також «Електронний щоденник».

Адміністратор даних може зареєструвати студента, якщо в того виникають певні проблеми, видалити його, зареєструвати викладача, додати нове місце проходження практики.

Під час виконання першого етапу курсової роботи на тему «Проходження практики», отримано досвід аналізу користувачів та ролей у інформаційній системі.

# 2 USE CASE UML ДІАГРАМИ

У даному етапі курсової роботи були створені use cases для кожного класу користувачів нашої інформаційної системи: для неавторизованого користувача, студента, викладача, адміністратора БД.

Неавторизований користувач може зареєструватись або ж пройти авторизацію.

Студент має такий функціонал: заповнення щоденника і перегляд щоденника.

Викладач має наступний функціонал: виставлення оцінок, перевірка щоденника.

Адміністратор БД має право змінювати інформацію про користувача, видаляти користувача, перевіряти щоденник, реєструвати нового викладача, додавати нове місце проходження практики

# 3 ДІАГРАМИ ПОСЛІДОВНОСТЕЙ

У даному етапі курсової роботи були створені sequence diagrams для кожного класу користувачів інформаційної системи. Також, під час виконання етапу курсової роботи, було поглиблено знання з аналізу користувачів, їх ролей та можливостей у ІС. Було встановлено, що неавторизований користувач може зареєструватись та авторизуватись. Студент може переглянути оцінки в своєму щоденнику, а також заповнити його. Викладач може перевірити щоденник користувача та виставити оцінки.

# 4 IDEF3

Всі діаграми побудовано за допомогою програмного забезпечення MS Visio.

1) Розглянемо діаграму нотації станів об’єкта «User» (зображена на рисунку 2.1) та відповідну діаграму потоків процесів (рис.3.1). Діаграма нотації станів має такі стани користувача: «User Unauth», «User reg», «User Auth», «User Droped». Отже, заходячи в систему, користувач знаходиться в стані «неавторизований користувач». Далі він має можливість зареєструватись в системі (перейти в стан «User Reg»), або ввійти в систему за раніше зареєстрованим в системі акаунтом («User Auth»), або так і залишитись неавторизованим («User Unauth»). Після реєстрації користувач може бути видалений з системи, а також може залогінитись. Після переходу в стан «User Auth», користувач також може бути видалений адміністратором з системи.

2) Розглянемо діаграму нотації станів об’єкта «Diary» (зображена на рисунку 2.2) та відповідну діаграму потоків процесів (рис.3.1). Діаграма нотації станів має такі стани: «Diary empty», «Diary edited», «Diary complete», «Diary deleted». Після того, як щоденник створений, він може бути відредагований (додавання студентом інформації, додавання оцінок викладачем) або видалений. «Diary complete» значить, що в щоденнику є 3 записи студента, 3 оцінки та остаточний результат. На всіх етапах щоденник може бути видалений.

# 5 МОДЕЛІ ЖИТТЄВИХ ЦИКЛІВ РОЗРОБКИ ОСНОВНИХ ПРОЦЕСІВ СИСТЕМИ

Процеси «Перевірка щоденника” і “Заповнення щоденника” розроблюються за спіральною моделлю життєвого циклу, бо можуть бути легко редаговані в ході використання системи, а також можуть бути покращеними в нових версіях.

Усі інші процеси розробляються з використанням RAD-моделі життєвого циклу для забезпечення швидкості розробки.

# 6 DFD

Розроблені в домашній роботі діаграми DFD складаються з наступних елементів:

а) зовнішні накопичувачі;

б) процеси обробки інформації;

в) внутрішні накопичувачі – використовуються для збереження інформації, яка отримується з процесів обробки інформації;

г) потік даних – стрілка, яка з’єднує між собою накопичувач і процес обробки, або навпаки.

Процеси між собою обмінюються інформацією лише через внутрішній накопичувач.

Побудова діаграм DFD більш високих рівнів полягає в декомпозиції кожного процесу попереднього рівня на складові підпроцеси, що виконують більш елементарні задачі.

При декомпозиції всі зовнішні, внутрішні накопичувачі, що були пов’язані з основним процесом, переносяться на даний рівень діаграми DFD.

В рамках виконання етапу курсової роботи отримано досвід розбиття бізнес-процесу на складові підпроцеси на прикладі процесу «Проходження практики».

На рівні DFD2 процес складається з:

- зовнішніх накопичувачів Teacher, Student, Organization for practice;

- внутрішніх накопичувачів diary, places, faculty, groups;

- процесів check mark, fill the diary, check diary, fill mark, analyze organization info, watch the map.

# 7 ERD

Розроблена в рамках виконання етапу курсової роботи діаграма «сутність-зв’язок» (ERD) для процесу «Оформлення візи» складається із:

а) сутностей (кількість – 8);

б) зв’язків.

Основний елемент діаграми ERD - це сутність: множина об’єктів зі спільними характеристиками, які називаються атрибутами. В роботі було використано зв’язки між сутностями наступних типів:

а) один до багатьох – реалізується імпортуванням ключа сутності до сутності, в якій відповідає хоча б двом об’єктам відповідає об’єкт з першої сутності;

б) багато до багатьох - реалізується за допомогою введення додаткової сутності, яка імпортує ключові атрибути пов’язаних сутностей та з’єднана з ними зв’язком «один до багатьох»;

Зв’язки характеризуються обов’язковістю з боку однієї сутності та необов’язковості з боку другої сутності.

Діаграма ERD, побудована в рамках виконання етапу курсової роботи, відповідає бізнес-логіці, яка була описана в попередніх етапах.

Що стосується зв'язків між сутностями, то вони розподілені наступним чином:

Сутність USERS має зв’язок типу «1-1» з сутністю STUDENT, це пов’язано з тим, що студент і є користувачем.

Сутність USERS має зв’язок типу «1-1» з сутністю TEACHER, це пов’язано з тим, що викладач і є користувачем.

Сутність STUDENT має зв’язок типу «1-N» з сутністю GROUP, це пов’язано з тим, що студент має лише одну групу, але в групі є багато студентів.

Сутність STUDENT має зв’язок типу «1-N» з сутністю FACULTY, це пов’язано з тим, що студент має лише один факультет, але на факультеті є багато студентів.

Сутність FACULTY має зв’язок типу «1-N» з сутністю GROUP, це пов’язано з тим, що група належить одному факультетові, але на факультеті існує багато груп.

Сутність STUDENT має зв’язок типу «1-N» з сутністю ORGANIZATION\_FOR\_PRACTICE, це пов’язано з тим, що студент може обрати одне місце проходження практики, але це місце можуть обрати декілька студентів.

Сутність STUDENT має зв’язок типу «1-N» з сутністю DIARY, це пов’язано з тим, що студент може мати лише один щоденник, але щоденник можуть мати багато студентів.

# 8 АРХІТЕКТУРИ СИСТЕМИ ДЛЯ ОСНОВНИХ ПРОЦЕСІВ ІС

В рамках виконання етапу було отримано практичні навички побудови архітектури системи для окремих блоків процесів системи на прикладі інформаційної системи «Проходження практики». Для обох блоків процесів даної інформаційної системи визначено архітектуру системи. Для блоку процесів «Авторизація користувача» в якості архітектури системи визначено «розподілене представлення даних» тому, що для даних операцій на стороні клієнта необхідна лише валідація даних, введених користувачем, а перевірка та/або збереження даних відбувається вже на сервері. Для блоку процесів «Перевірка щоденника» в якості архітектури системи визначено «розподілений застосунок».