ЗМІСТ

[1 РОЛІ КОРИСТУВАЧІВ В ІНФОРМАЦІЙНІЙ СИСТЕМІ 2](#_Toc473032895)

[2 USE CASE UML ДІАГРАМИ 3](#_Toc473032896)

[3 ДІАГРАМИ ПОСЛІДОВНОСТЕЙ 4](#_Toc473032897)

[4 IDEF3 5](#_Toc473032898)

[5 МОДЕЛІ ЖИТТЄВИХ ЦИКЛІВ РОЗРОБКИ ОСНОВНИХ ПРОЦЕСІВ СИСТЕМИ 6](#_Toc473032899)

[6 DFD 7](#_Toc473032900)

[7 ERD 8](#_Toc473032901)

[8 АРХІТЕКТУРИ СИСТЕМИ ДЛЯ ОСНОВНИХ ПРОЦЕСІВ ІС 9](#_Toc473032902)

# 1 РОЛІ КОРИСТУВАЧІВ В ІНФОРМАЦІЙНІЙ СИСТЕМІ

В ході виконання даного етапу курсової роботи було виділено чотири групи користувачів інформаційної системи «Здача зовнішнього незалежного оцінювання», а саме:

а) адміністратори інформаційної системи;

б) авторизований користувачі;

в) неавторизовані користувачі;

г) адміністратори даних.

Неавторизовані користувачі мають змогу лише зареєструватися та авторизуватися. Також вони бачать описову інформацію головної сторінки.

Авторизовані користувачі – люди, що планують пройти зовнішнє незалежне оцінювання. Вони мають змогу обирати предмети для проходження тестування або видаляти їх, переглядати графік проведення тестування, безпосередньо проходити тестування, та переглядати свої результати та список усіх своїх незалежних оцінювань.

Адміністратор даних має право видаляти користувача, додавати та видаляти предмети, з яких відбувається зовнішнє незалежне оцінювання, переглядати список людей, що здали оцінювання з конкретного предмету. А також додавати/видаляти тести та завдання.

# 2 USE CASE UML ДІАГРАМИ

При виконанні даного етапу курсової роботи біли побудовані Use Case діаграми для користувачів інформаційної системи, що описує бізнес-процес «Здача зовнішнього незалежного оцінювання».

Були розроблені Use Case діаграми для неавторизованих та авторизованих користувачів та для адміністратора даних.

Неавторизовані користувачі мають право лише зареєструватися або увійти під своїм логіном та паролем.

Авторизований користувач потрапляє у свій кабінет, де може продивлятись основну інформацію про ЗНО(предмети, дати, місця проведення). Додавати та видаляти тестування, які він планує пройти, передивлятись результати написаних тестів. Користувач не може змінювати результати своїх тестів.

Адміністратор даних також не може змінювати результати нічиїх тестів, але на відміну від користувача він може змінювати основну інформацію про проведення ЗНО, додавати та видаляти предмети, переглядати та змінювати тестові завдання.

# 3 ДІАГРАМИ ПОСЛІДОВНОСТЕЙ

Були побудовані діаграми послідовностей для неавторизованого користувача та авторизованого. У кожного на лінії життя різна кількість процесів. Неавторизований користувач має змогу зареєструватися або одразу авторизуватися. Авторизований користувач може виконувати послідовно різні дії пов’язані з переглядом основної, загальної інформації про тестування(розклад, предмети), та переглядом його особистої інформації(тести, які він проходив, результати). В деяких випадках потрібно зв’язуватись з базою даних через сервер, наприклад, щоб отримати результати тестувань після того, як вони були перевірені та внесені в базу.

# 4 IDEF3

Діаграма нотацій станів освітлює процес «з точки зору об’єкта», а діаграма потоків процесів «з точки зору спостерігача».

Обєкт «неавторизований користувач» може перейти в стани «зареєсртований користувач», а потім «залогінівшийся», або одразу перейти в стан «залогінівшийся». Після цього він перестає існувати як «неавторзований користувач». Обєкт «тест» після створення може одразу стати видаленим, або відредагованим(можна редагувати багато разів), а потім видаленим.

В ході процесу авторизації користувач або одразу логіниться, або спочатку реєструється, а потім логіниться.

Процес «створити новий тест» обов’язково включає в себе процеси «додати завдання», «встановити дату» та «встановити місце». Після цього можна або видалити тест, або редагувати. Редагування може включати або редагування завдань, та/або редагування дати/місця. Після цього можна або знов повернутись до редагування, або видалити тест.

# 5 МОДЕЛІ ЖИТТЄВИХ ЦИКЛІВ РОЗРОБКИ ОСНОВНИХ ПРОЦЕСІВ СИСТЕМИ

Каскадна модель була обрана для розробки процесів, що потребують високої ретельності та якості. До таких процесів, наприклад, відносяться ті, що потребують персональні дані користувача(наприклад, authorization), а також основні процеси, для яких дуже важливою є саме надійність роботи.

Спіральна модель була обрана для випадків, коли безпека та надійність не є критичними, для процесів, що потребують частого редагування(наприклад, schedule view).

Всі інші процеси розроблюються за спіральною моделлю, бо не мають вимог високої безпеки або якості, але в ході експлуатації можуть мати потребу в швидкій зміні або редагуванні, розроблюються ітераційно.

# 6 DFD

DFD нульового рівня для процесу «Здача зовнішнього незалежного оцінювання» складається з зовнішніх накопичувачі «адміністратор» та «користувач» та процесу «Здача ЗНО».

DFD першого рівня вже містить в собі чотири підпроцеси замість одного основного процесу: «створення зно», «здача тесту», «реєстрація на тестування» та «перегляд результатів».

Процеси між собою обмінюються інформацією лише через внутрішній накопичувач. Кожен з цих процесів в свою чергу також поділяється на підпроцеси. Суть полягає у тому, що ми один просторий процес розбиваємо на більш елементарні процеси, тим самим визначаємо все процеси системи, бачими все необхідні внутрішні накопичувачі.

# 7 ERD

Дана модель «сутність-зв’язок» містить шість основних сутностей, відповідно до основних процесів інформаційної системи: Users, Userwork, Useranswer, Test, Task, Subject.

Щоб уникнути зв’язку «багато до багатьох» були введені проміжні сутності. Наприклад: в одній роботі юзера може бути багато завдань(питань), а одне питання може бути у багатьох роботах. Була введена сутність «тест». Один тест має багато питань, конкретне питання може бути лише у певному тесті(один до багатьох), робота юзера може мати лише один тест, а кожен тест може бути у багатьох роботах(один до багатьох). Таким чином між сутностями реалізований зв'язок «один до багатьох».

# 8 АРХІТЕКТУРИ СИСТЕМИ ДЛЯ ОСНОВНИХ ПРОЦЕСІВ ІС

Було обрано архітектури для реалізації двох блоків інформаційної системи: архітектура «розподілене представлення даних» для блоку «Authorization» та архітектура «віддалене представлення даних» для блоку «Schedule view». Для блоку «Authorization» була обрана відповідна архітектура, бо на стороні клієнту проводиться лише валідація даних, та необхідність у високому рівні захисту інформації є критичною.

Для блоку «Schedule view» була обрана архітектура «віддалене представлення даних», бо функції, що мають змогу виконувати користувачі, зводяться до надсилання запитів(тобто прикладних функцій на клієнтській частині не буде), зможуть робити це пакетно, крім того вони матимуть доступ до вхідних/вихідних даних весь час(Data View знаходиться на стороні клієнта). Тобто у будь-який час можна передивлятись список тестувань та їх опис.