ЗМІСТ

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

[2 USE CASE UML ДІАГРАМИ 5](#_Toc470450045)

[3 ДІАГРАМИ ПОСЛІДОВНОСТЕЙ 6](#_Toc470450046)

[4 IDEF3 11](#_Toc470450047)

[5 МОДЕЛІ ЖИТТЄВИХ ЦИКЛІВ РОЗРОБКИ ОСНОВНИХ ПРОЦЕСІВ СИСТЕМИ 14](#_Toc470450048)

[6 DFD 15](#_Toc470450049)

[7 ERD 16](#_Toc470450050)

[8 АРХІТЕКТУРИ СИСТЕМИ ДЛЯ ОСНОВНИХ ПРОЦЕСІВ ІС 17](#_Toc470450051)

# 1 РОЛІ КОРИСТУВАЧІВ В ІНФОРМАЦІЙНІЙ СИСТЕМІ

В системі визначено наступних користувачів:

а) адміністратор ІС;

б) неавторизований користувач;

в) авторизований користувач;

г) адміністратор даних.

Користувачів кожного виду може бути більше ніж один, тобто маємо 4 базові групи користувачів.

Адміністратор ІС працює з налаштуваннями бекапу данних, оновлення даних з зовнішніх інтерфейсів, збереження та відправлення файлів на сервер.

Неавторизований користувач має наступні можливості:

- перегляд описової інформації про дисципліни, доступні в системі;

- реєстрація/авторизація в системі.

Вся множина авторизованих користувачів поділяється на 4 групи, тобто в системі, окрім базових груп користувачів (ролей), існує ще чотири групи (ролі):

а) викладач-лектор;

б) викладач-практик;

в) викладач лаборант;

г) студент.

Всі ці чотири групи в якості загальної області видимості даних мають власний акаунт. Викладачі практики та лаборанти в якості областей видимості даних мають:

- списки груп, в яких вони викладають з усіма оцінками за заняття або виконання робіт, пов`язаних лише з цим викладачем;

- розклад усіх своїх занять;

- здані цьому викладачу або перевірені цим викладачем роботи;

- повідомлення, надіслані студентам, або від студентів;

Відмінність областей видимості викладача-лектора від областей видимості інших викладачів, які викладають в рамках його дисципліни, полягає в наступному:

- списки груп лектора мають в собі оцінки за всі види робіт з даної дисципліни;

- лектор має доступ до всіх зданих або перевірених робіт;

Області видимості студента є наступними:

- список доступних дисциплін;

- список лабораторних та контрольних робіт з доступних дисциплін;

- залікові роботи з доступних дисциплін (за необхідністю);

- здані/перевірені роботи;

- списки всіх оцінок з роботами;

- повідомлення надіслані від викладачів, або викладачам.

Викладачі практики та лаборанти мають наступний список доступних операцій:

- додавання нової лабораторної роботи(для лаборанта) або нової контрольної роботи (для практика);

- видалення лабораторної роботи(для лаборанта) або контрольної роботи (для практика);

- перегляд та оцінювання зданих(надісланих) робіт;

- призначення додаткових занять;

- надсилання повідомлень студентам;

- перегляд/видалення повідомлень.

Викладач лектор має в доступі усі операції викладачів практики та лаборантів, які викладають в рамках його дисципліни. Також, лектор повинен підтверджувати операцію видалення або додавання роботи, яка була виконана викладачем практики або лаборантом.

Адміністратор даних підтверджує операції видалення або додавання роботи, виконані лектором.

Підтвердження відбувається на основі отриманих з зовнішнього інтерфейсу «Деканат» навчальних та робочих навчальних програм.

Також, адміністратор даних підтверджує реєстрацію в системі викладачів та студентів. Підтвердження відбувається також на основі отриманих з зовнішнього інтерфейсу «Деканат» даних. Також адміністратор може надіслати повідомлення авторизованому користувачу або забанити його за порушення правил поведінки в системі.

При виконанні етапу курсової роботи отримано досвід аналізу користувачів та ролей у інформаційній системі на прикладі інформаційної системи «Здача заліку».

# 2 USE CASE UML ДІАГРАМИ

Розроблені при виконанні даного етапу курсової роботи Use Case UML діаграми складаються з наступних елементів:

а) актор;

б) процес;

в) перехід.

Актор – це об’єкт ІС, що виконує первні процеси і поділяється на:

- master (адміністратори);

- slave (решта).

Процес описує інтерфейс, що надає користувачу певний функціонал.

Перехід вказує перехід користувача між інтерфейсами та взаємозв’язок між інтерфейсами.

Переходи бувають:

- вкладені;

- розширені;

Вкладений перехід від процесу А до процесу Б означає, що Б є вкладеним підпроцесом процесу А.

Розширений перехід від процесу А до процесу Б означає, що процес Б розширює функціонал А.

Різниця між вкладеним і розширеним переходами полягає в тому, що для вкладених їх вершина є абстракцією, що реалізується через них. Для розширених переходів вершина існує як окремий модуль.

Діаграми, побудовані під час виконання даного завдання, для користувачів-не-адміністраторів мають обов'язкову точку входу – авторизацію. Всі інші процеси залежать від конкретного користувача.

При виконанні домашньої роботи отримано досвід побудови Use Case UML діаграм для користувачів інформаційної системи, що описує бізнес-процес «Здача заліку».

# 3 ДІАГРАМИ ПОСЛІДОВНОСТЕЙ

Діаграма послідовностей для неавторизованого користувача має три паралельні послідовності.

Перша послідовність показує дії неавторизованого користувача без реєстрації та входу в систему під конкретними даними. При загальному вході в систему користувач передає в процес temp\_user свою IP-адресу та час входу. Система за допомогою цього процесу тимчасово реєструє даного користувача, одночасно передаючи дані адміністратору AdminD для контролю за тимчасовими користувачами. Далі користувач без самостійної реєстрації або входу в систему може лише вибрати предмет і подивитись про нього інформацію, передавши в процес Subject Description назву предмету і отримавши описову інформацію про предмет.

Друга послідовність показує взаємодію об’єктів системи при самостійній реєстрації неавторизованого користувача. Користувач передає в процес Reg свій логін, пароль, електронну пошту, ПІБ та роль. Процес посилає ці дані до AdminD для верифікації ПІБ та ролі та посилає повідомлення користувачу на електронну пошту для її підтвердження. Після того як користувач підтвердив адрес електронної пошти, перейшовши за адресою, вказаною в листі, інформація про активацію переходить в процес Reg, який цю інформацію передає до AdminD. AdminD запитує і отримує інформацію деканату, з DataServer, для верифікації нового профілю. Після того, як AdminD верифікує створений в системі новий профіль користувача, користувачу передається статусна інформація успішності або не успішності верифікації його профілю. У разі успішності дані профілю зберігаються на DataServer.

Третя послідовність показує взаємодію об’єктів системи при вході неавторизованого користувача під конкретними даними. Користувач передає в процес login свої логін та пароль. В процесі відбувається верифікація цих даних. У разі неуспішної верифікації (логін і пароль не співпадають), користувач вводить ці дані повторно. У разі успішної верифікації і в залежності від ролі користувача в системі, користувач входить в систему або як студент, або як лектор, або як асистент (практик або лаборант), або як AdminIS. Від процесу login користувач отримує статусну інформацію про результат самостійного входу в систему.

Надалі будемо вважати, що всі процеси, які потребують завантаження або відправлення даних (таких, як листи, звіти, розклади занять тощо) завантажуються або вивантажуються відповідно з DataServer.

Діаграми студента має наступні послідовності:

- послідовності, пов’язані з розкладом занять;

- послідовності, пов’язані з поштою;

- послідовності, пов’язані з журналом оцінок;

- послідовності, пов’язані з різноманітними завданнями.

Студент має можливість подивитись розклад двох видів: розклад всіх своїх занять та розклад занять з конкретної дисципліни. Для того, щоб подивитись весь свій розклад занять, користувач передає в процес View schedule of all my subjects інформацію про себе та отримує відповідний розклад. Для отримання розкладу занять з конкретної дисципліни користувач передає в процес View schedule of subject ще й назву дисципліни, розклад якої цікавить користувача.

Студент має можливість роботи з поштою: читання повідомлень та написання запитань викладачам. Для написання запитань викладачам користувач передає в процес Ask a question інформацію про себе та саме тіло листа-запитання, в яке входить текст листа, назва дисципліни, якої стосується запитання, файли, які прикріплені до листа. Як результат, користувач отримує статусну інформацію про успішність відправки листа. Для читання повідомлення користувач передає в процес Read letter інформацію про себе та ідентифікатор листа, який користувач хоче прочитати. В результаті користувач отримує повідомлення.

Студент має можливість дивитися свої оцінки: всі оцінки чи оцінки з конкретної дисципліни. Схема взаємодії об’єктів аналогічна схемі взаємодії об’єктів у випадку роботи з розкладом занять, якщо замінити розклад занять на оцінки, а процеси View schedule of all my subjects та View schedule of subject на процеси All marks та Subject marks відповідно.

Студент має можливість виконувати різноманітні завдання. Для вибору завдання користувач передає в процес Choose work інформацію про себе, тип завдання (лабораторна робота, практичне завдання, контрольна робота, тест, залікова контрольна робота) та ідентифікатор завдання та отримує саме завдання. Для відправлення результату на перевірку користувач відправляє у процес Send result окрім вище вказаних параметрів ще й результат виконання завдання. Процес Send result, після завершення перевірки результату виконання викладачем в рамках цього процесу, відправляє перевірений результат в процес користувача, призначений для аналізу помилок Analysis of chosen work. Після отримання перевіреної роботи, процес Analysis of chosen work відправляє статусну інформацію в процес Send result про успішне отримання перевіреної роботи, а процес Send result відправляє статусну інформацію користувачу про закінчення перевірки роботи. Користувач після цього може проаналізувати помилки в перевіреному результаті виконаного завдання, відправивши в процес Analysis of chosen work той самий набір параметрів, що й у процес Choose work та отримавши в результаті перевірений результат виконаного завдання.

Лектор має дещо ширші можливості, ніж студент, хоча лектор не має доступу до деяких процесів, які доступні студенту.

Наприклад, процес роботи з поштою Ask a question у лектора замінюється на процес Create a letter, який, на відміну від попереднього процесу, дає можливість лектору вибрати, кому зі студентів відправити повідомлення.

Робота з розкладом занять та завданнями доповнюється однотипними за принципом дії процесам New or Del lesson та New or Del work відповідно. Процеси дають можливість лектору додати нове заняття та завдання або видалити заняття (відмінити його) та завдання. Для того щоб видалити або додати заняття, лектор відправляє в процес New or Del lesson інформацію про себе, час заняття, інформацію про заняття та тип запиту (видалення або додавання). Цей процес передає надану інформацію на погодження до AdminD. AdminD після верифікації цього питання надає наступну інформацію процесу New or Del lesson: час заняття та інформацію про нього і відповідь так чи ні на цей запит. У випадку позитивної верифікації питання інформація про заняття разом з часом проведення передається в процес View schedule of all my subjects для подальшої обробки даного питання. Від процесу New or Del lesson до користувача надходить статусна інформація про успішність верифікації питання. Після обробки питання, процес View schedule of all my subjects надає користувачу оновлений розклад занять.

За аналогічним принципом відбувається взаємодія об’єктів відносно процесу New or Del work, тільки замість заняття розглядається завдання та до атрибутів додається ще й тип завдання.

В роботі з журналом окрім процесів, які доступні студенту, додається процес Group subject marks, який надає лектору інформацію про оцінки конкретної групи з його предмету. На вхід від лектора до процесу надходить інформація про лектора, назва предмету та назва групи.

Робота лектора над завданнями відрізняється від роботи студента над завданнями. За допомогою процесу Choose result лектор може переглянути результат виконання кимось конкретного завдання (на вхід процесу надходить інформація про лектора, тип завдання, ідентифікатор завдання, інформація про студента) та перевірити його, якщо він є не перевіреним. Тоді за допомогою процесу Send checked work лектор може відправити перевірений результат виконання конкретного завдання якимось студентом.(на вхід процесу надходить інформація про лектора, тип завдання, ідентифікатор завдання, тіло перевіреного результату, інформація про студента). Як результат, лектор отримує статусну інформацію про відправлення перевіреного результату.

Асистент відрізняється від лектора тим, що він має вплив лише на завдання конкретного типу (лаборант – на лабораторні роботи, практик – на практичні та контрольні роботи), має доступ до оцінок лише цих робіт та контролюється лектором (на відміну від асистента, лектор контролюється адміністратором AdminD). Тому діаграма послідовностей асистента подібна до діаграми послідовностей лектора.

Діаграма послідовностей AdminIS має три паралельні послідовності. За допомогою першої послідовності він встановлює налаштування бекапу даних(процес Backup config, на вхід процесу поступають налаштування бекапу даних). За допомогою другої послідовності він встановлює налаштування вмикання та вимикання серверу (Server on off config, на вхід процесу поступають налаштування вмикання та вимикання серверу). За допомогою третьої послідовності він встановлює налаштування оновлення даних (Update config, на вхід процесу поступають налаштування оновлення даних).

# 4 IDEF3

а) Діаграма ієрархії процесів, що зображена на рисунку 2.1, має чотири рівні ієрархії. На першому рівні знаходиться головний процес (Здача заліку). На другому рівні цей процес розбивається на три: «Subject view», «Marks view», «Auth». На третьому рівні всі три процеси декомпонуються. Процес «Subject view» декомпонується на такі процеси: «New subject», «Works view», «Subject description edit», «Del subject», «Subject description view». Процес «Marks view» декомпонується на такі процеси «All marks», «Subject marks view», «Group subject marks view». Процес «Auth» декомпонується на такі процеси: «Enter in IS», «Reg», «Login», «temp\_user», «Drop user». На четвертому рівні декомпонується лише процес «Works view». Він декомпонується на такі процеси: «New work», «Edit work», «Choose work for doing», «Send a result», «Choose work for checking», «Send a checked work», «Analyze checked work», «Close for redoing», «Redoing of work», «Del work». Діаграма виконана за допомогою програмного забезпечення PowerDesigner

б) У відповідних нижче згаданих парах діаграм існує відповідність між назвами процесів та назвами станів, в які відповідні процеси переводять відповідні об’єкти. Всі згадані нижче діаграми виконані за допомогою програмного забезпечення MS Visio.

1) Розглянемо діаграму нотації станів об’єкта «користувач» (зображена на рисунку 3.1) та діаграму потоків процеса «авторизація» (зображена на рисунку 4.1). Діаграма нотації станів має п’ять станів: «користувач неавторизований», «користувач зареєстрований», «користувач залогінився», «користувач тимчасово зареєстрований», «користувач видалений з системи». Відповідно до обох вище згаданих діаграм, користувач заходить в систему, знаходиться в стані «неавторизований користувач». Потім, такий користувач має можливість самостійно зареєструватись в системі (перейти в стан «зареєстрований користувач»), або ввійти в системі за раніше зареєстрованим в системі аккаунтом (перейти в стан «користувач залогінився»), або бути тимчасово зареєстрованим в системі (перейти в стан «користувач тимчасово зареєстрований»). Після самостійної реєстрації користувач може бути видалений з системи, якщо він не має права доступу до системи відповідно до інформації, яку має AdminD, або користувач може залогінитись. Після переходу в стан «користувач тимчасово зареєстрований» або «користувач залогінився», користувач також може бути видалений користувачем AdminD з системи відповідно до інформації, яка в нього є.

2) Розглянемо діаграму нотації станів об’єкта «дисципліна» (зображена на рисунку 3.2) та діаграму потоків процеса «Subject view» (зображена на рисунку 4.2). Діаграма нотації станів має чотири стани: «дисципліна не створена», «нова дисципліна», «відредагована дисципліна», «видалена дисципліна». Відповідно до обох вище згаданих діаграм, дисципліна створюється відповідним процесом. Потім дисципліна може бути відредагована (додавання або видаленням робіт, зміною опису дисципліни) або видалена. Після переходу в стан «відредагована дисципліна», дисципліна може бути видалена. Створення або видалення дисципліни відбувається викладачем-лектором даної дисципліни при погодженні користувача AdminD.

3) Розглянемо діаграму нотації станів об’єкта «робота» (зображена на рисунку 3.3) та діаграму потоків процеса «Works view» (зображена на рисунку 4.3). Діаграма нотації станів має десять станів: «робота не створена», «нова робота», «відредагована робота», «вибрана робота для виконання», «виконана робота», «вибрана робота для перевірки», «перевірена робота», «робота проаналізована», «робота без можливості перевиконання», «робота видалена». Відповідно до обох вище згаданих діаграм, робота створюється відповідним процесом. Потім робота може бути або відредагована, або видалена, або вибрана для виконання. Після відредагування робота може бути видалена або вибрана для виконання. Після виконання роботи студент відправляє результат і робота переходить в стан виконаної. Після цього викладач обирає цю роботу для перевірки. Після перевірки він відправляє перевірену роботу і сама робота переходить в стан перевіреної. Після цього студент має можливість проаналізувати її перевірку. Як тільки він це зробить, робота переходить в стан проаналізованої. Після переходу в цей стан робота може бути перевиконана студентом (робота переходить знову в стан обраної для виконання і проходить вище згаданий ланцюг до стану «проаналізована робота») або може бути переведена в стан «робота без можливості перевиконання», що унеможливлює її перевиконання студентом (одразу це робиться для виконаної залікової роботи у разі набору допустимої кількості балів, для всіх інших видів робіт це рішення залишається на розсуд викладачу). Після цього, робота може бути видалена. Видалення роботи відбувається відповідним асистентом з погодження лектора або лектором з погодження користувача AdminD.

4) Розглянемо діаграму нотації станів об’єкта «оцінка» (зображена на рисунку 3.4) та діаграму потоків процеса «Works view» (зображена на рисунку 4.3). Діаграма нотації станів має п’ять станів: «оцінка не створена», «оцінка створена», «оцінка відредагована», «оцінка без можливості редагування», «оцінка видалена». Відповідно до обох вище згаданих діаграм, оцінка створюється при створенні роботи (вона дорівнює 0). Після створення оцінка може бути відредагована (після виконання або перевиконання студентом роботи, а саме – після надсилання перевіреної роботи викладачем) або видалена. Після того, як оцінка є відредагованою, оцінка може перейти в стан «оцінка без можливості редагування» (після заборони перевиконання роботи, якій відповідає дана оцінка; після першого редагування це робиться для виконаної залікової роботи у разі набору допустимої кількості балів, для всіх інших видів робіт це рішення залишається на розсуд викладачу). Після цього, робота може бути видалена. Видалення відбувається у випадку якщо відбувається видалення відповідної роботи.

# 5 МОДЕЛІ ЖИТТЄВИХ ЦИКЛІВ РОЗРОБКИ ОСНОВНИХ ПРОЦЕСІВ СИСТЕМИ

Процеси Reg, Login, temp\_user, Drop user розроблюються за каскадною моделлю життєвого циклу, бо повинні мати високий рівень безпеки, бо включають в себе роботу з (аутентифікаційними) даними користувача. Процеси New subject, New work, Edit work, Del work, Del subject розроблюються за каскадною моделлю життєвого циклу, бо повинні мати високу якість розробки, оскільки є основотворними процесами в системі. Процеси Send a result/checked work розроблюються за каскадною моделлю життєвого циклу, бо повинні мати високий рівень безпеки, оскільки працюють з завантаженням на сервер результатів роботи студента та результатів перевірки викладачем.

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

# 6 DFD

Розроблені в домашній роботі діаграми DFD складаються з наступних елементів:

а) зовнішні накопичувачі (джерела інформації);

б) процеси обробки інформації;

в) внутрішні накопичувачі (джерела інформації) – 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.

Внутрішні накопичувачі необов’язково відображаються на діаграмах DFD нульового рівня. На ньому обов’язково зображуються зовнішні накопичувачі, головний процес обробки інформації та потоки від накопичувачів до процесу.

Побудова діаграм DFD більш високих рівнів полягає в декомпозиції кожного процесу попереднього рівня на складові підпроцеси, що виконують більш елементарні задачі. При цьому, на таких діаграмах вже повинні бути внутрішні накопичувачі.

Декомпозицію зроблено за допомогою побудови дерева процесів. При цьому всі зовнішні, внутрішні накопичувачі, що були пов’язані з основним процесом, переносяться на даний рівень діаграми DFD. На кожному наступному рівні на деяких діаграмах з’являються нові внутрішні накопичувачі.

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

# 7 ERD

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

а) сутностей (кількість – 31);

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»;

в) один до одного - реалізується за допомогою введення додаткової сутності, яка імпортує ключові атрибути пов’язаних сутностей та з’єднана з ними зв’язком «один до багатьох», але для сутності, яка знаходиться зі сторони «багато» реалізуються програмно алгоритми перевірки на відсутність дублікатів об’єктів сутності, яка стоїть зі сторони «один» (тому, фактично сторона «багато» за сенсом є аналогічною стороні «один»).

Окремі зв’язки характеризуються обов’язковістю з боку однієї сутності та необов’язковості з боку другої сутності. Присутні також зв’язки, які характеризуються обов’язковістю з обох боків.

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

При виконанні домашньої роботи закріплено досвід визначення сутностей та зв’язків між ними і побудови діаграми «сутність-зв’язок» на прикладі процесу «Здача заліку».

# 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 «Здача заліку». 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 «Перегляд дисциплін» в якості архітектури системи визначено «розподілений застосунок» тому, що користувач може змінювати на свій розсуд графічне представлення необхідної йому інформації навчального процесу та формувати список дисциплін та робіт за допомогою фільтрів, а також – користувач потребує можливості пакетно відправляти дані (файли звітів).