Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Кафедра прикладної математики

ЗВІТ

ПРО ВИКОНАННЯ VІ ЕТАПУ КУРСОВОЇ РОБОТИ

з дисципліни “Бази даних та інформаційні системи”

на тему: Здача заліку

Студента ІV курсу, групи КМ-32 напряму підготовки 6.040301 – прикладна математика

ЖУКА І. С.

Викладач

ТЕРЕЩЕНКО І.О.

Оцінка: \_\_\_ балів

Київ – 2016

ЗМІСТ

[ВСТУП 3](#_Toc466935122)

[1 ОПИС ЗАВДАННЯ 4](#_Toc466935123)

[2 ОПИС РЕЗУЛЬТАТІВ 5](#_Toc466935124)

[ВИСНОВКИ 15](#_Toc466935125)

[ПЕРЕЛІК ПОСИЛАНЬ 16](#_Toc466935126)

[Додаток А 17](#_Toc466935127)

[Додаток Б 19](#_Toc466935128)

# ВСТУП

Аналіз бізнес-процесу організаційної системи є важливим для побудови коректної інформаційної системи. Одним із способів такого аналізу є розбиття бізнес-процесу на складові та побудова дерева процесів на основі отриманих складових. Цей спосіб допомагає визначити які процеси треба реалізувати в інформаційній системі.

Інформаційна система – це комплекс інформаційних технологій, що реалізують бізнес процеси. Однією з технологій, що використовується для моделювання бізнес-процесів, є побудова діаграм DFD, які є графічним представленням потоків даних між процесами бізнес-процесу.

В рамках виконання даного етапу курсової роботи потрібно провести аналіз бізнес-процесу «Здача заліку» шляхом побудови діаграм DFD0, DFD1, DFD2.

# 1 ОПИС ЗАВДАННЯ

Завдання - описати бізнес-процес «Здача заліку», а саме, побудувати:

а) DFD0;

б) DFD1;

в) DFD2.

# 2 ОПИС РЕЗУЛЬТАТІВ

Виходячи з завдання, було побудовано:

а) DFD0 (додаток А);

б) DFD1 (2 підпроцеси для DFD0, додаток А);

в) DFD2 (додаток Б).

Потоки Read означають, що склад внутрішнього накопичувача повністю зчитується в процес, до якого направлена стрілочка з відповідною назвою.

Розглянемо діаграми DFD1, DFD2.

Розглянемо спочатку процес авторизації Auth та його підпроцеси (рисунки А.2 та Б.1).

З зовнішньої сутності Deanery (Деканат) отримується актуальна інформація про студентів, викладачів, навчальні дисципліни. Ця інформація запитується адміністратором через процес Create/update deanery information у зовнішньої сутності Deanery потоком subjects, students and teachers query. Зовнішня сутність Deanery відправляє в процес інформацію про студентів, вчителів, дисципліни потоком subjects, students and teachers info. Результат процесу: оновлення даних про викладачів (потік Updated teachers info), студентів (потік Updated students info), дисципліни (потік Updated subject deanery info) та взаємозв’язок між даними трьома об’єктами (потік Updated subjects, teachers and students info). Потік Updated students info має наступну структуру:

а) student\_name - тип string, ПІБ студента;

б) student\_nsc – тип string, номер студентського квитка;

в) student\_ng – тип string, номер групи студента;

г) idcf – тип string, ідентифікатор кафедри;

д) idfс – тип string, ідентифікатор факультету.

Внутрішній накопичувач Student info має ту саму структуру, що й потік Updated students info, причому всі атрибути окрім першого, є ключем сутності.

Потік Updated teachers info має наступну структуру:

а) teacher\_name – тип string, ПІБ вчителя;

б) teacher \_np – тип string, номер паспорту;

в) idcf – тип string, ідентифікатор кафедри;

г) idfс – тип string, ідентифікатор факультету;

Внутрішній накопичувач Teacher info має ту саму структуру, що й потік Updated teachers info, причому всі атрибути окрім першого, є ключем сутності.

Потік Updated subject deanery info має наступну структуру:

а) sbjct\_name – тип string, назва предмету;

б) sbjct\_сode – тип string, код дисципліни;

в) idcf – тип string, ідентифікатор кафедри;

г) idfс – тип string, ідентифікатор факультету;

д) sbjct\_dscrptn\_info – тип text, опис предмету.

Внутрішній накопичувач Subject deanery info має ту саму структуру, що й потік Updated subject deanery info, причому всі атрибути окрім першого і останнього, є ключем сутності.

Потік Updated subjects, teachers and students info має наступну структуру:

а) sbjct\_сode – тип string, код дисципліни;

б) ng\_idcf – тип string, ідентифікатор кафедри групи;

в) ng\_idfс – тип string, ідентифікатор факультету групи;

г) student\_ng – тип string, номер групи студента;

д) teacher\_name – тип string, ПІБ вчителя;

е) teacher\_np – тип string, номер паспорту вчителя;

ж) teacher\_idcf – тип string, ідентифікатор кафедри вчителя;

з) teacher\_idfс – тип string, ідентифікатор факультету вчителя;

к) teacher\_role – тип string, роль вчителя в предметі.

Внутрішній накопичувач Subjects, teachers and students info має ту саму структуру, що й потік Updated subjects, teachers and students info, причому всі атрибути окрім останнього, є ключем сутності.

Отримана інформація використовується при реєстрації нових студентів, викладачів, навчальних дисциплін або робіт з навчальних дисциплін для перевірки та підтвердження або відхилення відповідної реєстрації. Після підтвердження реєстрації адміністратором нові об’єкти (інформація про них) відповідно зберігаються на наступних внутрішніх накопичувачах:

а) Student – інформація про студентів:

1) student\_login – тип string, логін користувача-студента, ключ сутності;

2) student\_password – тип string, пароль користувача-студента;

3) student\_name – тип string, ПІБ студента;

4) student\_nsc – тип string, номер студентського квитка;

5) student\_ng – тип string, номер групи студента;

6) idcf – тип string, ідентифікатор кафедри;

7) idfс – тип string, ідентифікатор факультету;

8) student\_status - тип string, статус студента.

б) Teacher – інформація про викладачів:

1) teacher\_login – тип string, логін користувача-вчителя, ключ сутності;

2) teacher\_password – тип string, пароль користувача-вчителя;

3) idcf – тип string, ідентифікатор кафедри;

4) idfс – тип string, ідентифікатор факультету;

5) teacher\_name – тип string, ПІБ вчителя;

6) teacher \_np – тип string, номер паспорту;

7) teacher\_status - тип string, статус викладача.

в) Subject info – інформація про дисципліну:

1) sbjct\_id – тип string, ідентифікатор предмету, ключ сутності (створюється системою при створенні нової дисципліни);

2) sbjct\_name – тип string, назва предмету, ключ сутності;

3) sbjct\_teacher\_lector\_login – тип string, логін викладача-лектора;

4) sbjct\_lab\_teacher\_login – тип string, логін викладача-лаборанта;

5) sbjct\_practic\_teacher\_login – тип string, логін викладача практики;

5) sbjct\_grouplist – список елементів типу string, кожен з яких являє собою номер групи, для якої доступна ця дисципліна;

7) sbjct\_dscrptn\_info – тип text, опис предмету;

8) sbjct\_status – тип string, статус дисципліни.

г) Work – інформація про роботи з дисципліни:

1) sbjct\_id – тип string, ідентифікатор предмету, ключ сутності;

2) work \_id – тип string, ідентифікатор роботи, ключ сутності (створюється системою при створенні нової роботи);

3) work\_type – тип string, означає власне тип роботи (тест, лабораторна робота, контрольна робота, залікова контрольна);

4) work\_info - тип text, опис роботи;

5) work\_status – тип string, статус роботи.

Реєстрація користувачів відбувається наступним чином. З зовнішнього інтерфейсу User, який являє собою користувача системи, відправляється інформація, яку новий користувач вводить при реєстрації, за допомогою потоку reg info у процес Auth, підпроцес Reg. До цього процесу підключається адміністратор за допомогою потоку administrating з внутрішньої сутності Admins, що являє собою авторизованих адміністраторів. Також в процес завантажується інформація деканату про студентах з накопичувача Student info, та вчителів з накопичувача Teacher info. Після перевірки та підтвердження або відхилення реєстрації адміністратором користувачу (зовнішньому інтерфейсу) посилається інформація про статус реєстрації (потік reg status). У разі успішної реєстрації інформація про користувача, відповідно до ролі нового користувача в системі, зберігається у внутрішній накопичувач Student (для студентів, потік registered student) або Teacher (для вчителів, потік registered teacher). Потоки registered student та registered teacher мають ту саму структуру, що й відповідні їм внутрішні накопичувачі.

Після того, як в рамках процесу авторизації (процес Auth, підпроцес Reg) були зареєстровані нові користувачі, ці користувачі можуть увійти в систему. З зовнішнього інтерфейсу User до процесу Auth (підпроцес Login) передається логін та пароль (login, pass). Також у вказаний процес та підпроцес завантажується інформація з внутрішніх накопичувачів Student, Teacher, Admin (інформація про адміністраторів). Після здійснення спроби входу, користувачу (зовнішньому інтерфейсу) надсилається інформація про статус операції входу в систему (чи є вхід успішним, чи ні). Якщо дані, які ввів користувач, існують в системі (існують на вказаних накопичувачах), то змінюється статус відповідного користувача. Зміна статусу відбувається за допомогою надсилання на відповідний внутрішній накопичувач логіну користувача (як унікального ключа для пошуку та зміни даних відповідного користувача на внутрішньому накопичувачі) та статусу користувача (який дорівнює значенню «authorized» (авторизований)) в рамках потоку authorized\_<user>, де <user> = student або teacher, або admin відповідно.

Всі користувачі при вході в систему тимчасово реєструються. Відбувається це за допомогою надсилання із зовнішньої сутності User у процес temp\_user даних ip (ip-адреса користувача) та time (час входу в систему). Результатом процесу temp\_user (тимчасова реєстрація користувача) є наступна структура даних (структура передається потоком даних reg temp user):

- temp\_user\_login - тип string, логін тимчасового користувача (його ip-адреса);

- temp\_user\_time - тип date, час входу користувача в систему.

Структура зберігається на внутрішньому накопичувачі temp\_user. Інших елементів у структурі накопичувача немає. Ключем сутності є temp\_user\_login.

Така структура даних являє собою тимчасового користувача системи. Цей користувач може зареєструватись або залогінитись під своїм логіном та паролем, або переглянути інформацію про навчальні дисципліни. Вхід тимчасового користувача в відповідні процеси відбувається за допомогою надсилання інформації в рамках потоків logining, registering, temp\_user info, структура яких співпадає зі структурою внутрішнього накопичувача temp\_user. Процеси входу тимчасового користувача під своїми даними Login та реєстрації тимчасового користувача під своїми даними Reg описано вище. Процес перегляду інформації про навчальні дисципліни Subject description view буде описано нижче.

Користувачі можуть бути видалені з системи або їх профіль може бути відредагований у разі зміни про них інформації, яка стосується навчального процесу. Їх може видалити адміністратор системи. Опис процесу має наступний вигляд. У процес видалення (зміни інформації про) користувача з системи change/drop user надходить інформація з внутрішньої сутності Admin (що являє собою вхід адміністратора в процес, потік даних administrating). Завантажується інформація з накопичувачів, які зберігають інформацію деканата про студентів та викладачів. В процес паралельно з цим завантажується вся інформація про викладачів та студентів, які є в системі. Результатом процесу є:

- зміна статусної інформації користувачів, які повинні бути видалені з системи або інформація про яких повинна бути змінена згідно інформації з деканату (потоки типу change/drop user info, структура таких потоків – це набір структур вигляду [<user>\_login; <user>\_info; <user>\_status], де <user>\_info – всі інші поля інформації про відповідного користувача, окрім паролю, <user>\_status – статус відповідного користувача (дорівнює changed у випадку зміни інформації або dropped у випадку видалення користувача з системи));

- надсилання відповідним користувачам інформації про видалення з системи або зміни інформації (потік drop status, що направлений до зовнішньої сутності User).

Для адміністратора процеси входу в систему та реєстрації співпадають з процесами, розглянутими вище для звичайних користувачів.

Внутрішній накопичувач Admin має наступну структури:

- admin\_login – тип string, логін користувача-адміністратора, ключ сутності (внутрішнього накопичувача Admin);

- admin\_password – тип string, пароль користувача-адміністратора;

- admin\_status - тип string, статус адмінітратора, у випадку створення дорівнює created, у випадку видалення дорівнює dropped.

Процес Auth з підпроцесами повністю розглянуто. Розглянемо тепер процес Subject view (рисунок А.2 та рисунок Б.2).

Викладач може створити нову дисципліну. Умовно це можна описати наступним чином. Викладач входить в процес створення нової дисципліни New subject (робиться це передаванням даних викладача в рамках потоку teacher info, який складається з логіну вчителя та кафедри й факультету викладача, з внутрішнього накопичувача Teacher). При створенні нової дисципліни викладач вводить необхідну інформацію про дисципліну, також може додати роботи, пов’язані з цією дисципліною (це робиться передавання інформації з зовнішньої сутності User в рамках потоку teacher new subject info). Також викладач додає інформацію про навчальні групи, до яких відноситься ця дисципліна (для цього з внутрішнього накопичувача Student завантажується інформація про всіх студентів, представляючись викладачу у вигляді списку груп). Після цього, в процес входить адміністратор (потік даних administrating від внутрішнього накопичувача Admin). В процес завантажується інформація деканату про дисципліну з відповідних внутрішніх накопичувачів. Результатом процесу є:

- створення нової дисципліни (зберігається новий об’єкт в накопичувач Subject info за допомогою потоку даних info of created subjects, структура якого співпадає зі структурою відповідного йому внутрішнього накопичувача);

- створення журналу дисципліни(створення множини нових об’єктів на внутрішньому накопичувачі Mark (об’єкти накопичувача - оцінки));

- створення нових робіт (створення нових об’єктів на внутрішньому накопичувачі Work).

Викладач може передивлятись, створювати та видаляти роботи (в залежності від своєї ролі, було описано в попередніх етапах курсової роботи), перевіряти звіти студентів з виконаних робіт, студент може передивлятись роботи, які йому треба виконати для здачі заліку з предмету, відправляти звіти з виконаних робіт на перевірку. Все це можна робити в рамках процесу перегляду робіт Work view.

Для перегляду робіт, відправлення звітів студент входить в цей процес шляхом відправлення даних студента з внутрішнього накопичувача Student (потік даних student info). Також для цього завантажується інформація про дисципліну (внутрішній накопичувач Subject info, потік subject info), інформація про роботи (внутрішній накопичувач Work) та переглянуті роботи (внутрішній накопичувач Reviewed work). Перегляд роботи зберігається у вигляді наступної структури (потік даних reviewed work info) (таку структуру має внутрішній накопичувач Reviewed work): student\_login, sbjct\_id, work\_id, time (час перегляду роботи, в процес передається з зовнішньої сутності User) (всі елементи структури є ключем сутності). Звіт з виконаної роботи відправляється у процес з зовнішньої сутності User, зберігається звіт у вигляді наступної структури (потік даних new report) (таку структуру має внутрішній накопичувач Report): teacher\_login (логін викладача, який перевіряє роботу), student\_login, sbjct\_id, work\_id, (перші чотири елементи структури є ключем сутності внутрішнього накопичувача Report), report\_body (тип object, тіло звіту з роботи), report\_status (тип string, статус звіту з роботи).

Для перегляду робіт, створення нових та видалення старих робіт, перевірки звітів, відправлення перевірених звітів вчитель входить в цей процес шляхом відправлення даних вчителя з внутрішнього накопичувача Teacher (потік даних teacher info). Також для цього завантажується інформація про дисципліну (внутрішній накопичувач Subject info, потік subject info), інформація про роботи (внутрішній накопичувач Work), інформація про звіти (внутрішній накопичувач Report), інформація про оцінки (внутрішній накопичувач Mark) та інформацію про студентів (внутрішній накопичувач Student). Перевірений звіт з виконаної роботи відправляється у процес з зовнішньої сутності User, зберігається шляхом зміни у, відповідному звіту роботи, об’єкті внутрішнього накопичувача Report статусу на checked (при перевірці перевиконаної роботи - rechecked) (потік даних checked report), і зміни у, відповідному роботі, об’єкті внутрішнього накопичувача Mark значення оцінки на те, яку оцінку виставлено студенту за виконану роботу (потік даних checked work mark).

Внутрішній накопичувач Mark має наступну структуру:

- mark\_id – тип string, ідентифікатор оцінки, ключ сутності (створюється системою при створенні нової оцінки);

- student\_login – ключ сутності;

- sbjct\_id – ключ сутності;

- work\_id – ключ сутності;

- mark\_value – тип int, числове значення оцінки;

- mark\_status – тип string, статус оцінки.

Викладачі та студенти можуть переглядати оцінки. Для того, щоб студент переглянув оцінки, в процес перегляду оцінок Mark view передаються дані про нього з відповідного внутрішнього накопичувача, а також передаються дані з накопичувача Subject info, Mark, Report (потік даних checked reports, перевірені звіти). Результатом процесу. Результатом є зміна статусу відповідної переглянутою оцінки на reviewed (при перегляді оцінки студент може побачити, які помилки допустив при виконанні роботи) (потік даних reviewed marks). Для того, щоб викладач переглянув оцінки, в процес перегляду оцінок Mark view передаються дані про нього з відповідного внутрішнього накопичувача, а також передаються дані з накопичувача Subject info, Mark, Report, Student (для того, щоб переглянути оцінки як по групам, так і по окремому студенту). З цієї точки зору результатом процесу може бути зміна статусу відповідних переглянутих оцінок на closed (означає, що дані перевірені роботи не перездаються).

Викладач може змінювати опис дисципліни. Для цього, в процес редагування опису дисципліни Subject description edit передається інформація про викладача з внутрішнього накопичувача Teacher та інформація про дисципліну з Subject info. Паралельно з цим завантажується інформація деканату про дисципліну з відповідного внутрішнього накопичувача. Результатом є змінений опис відповідної дисципліни, обраної викладачем, в якій він змінив опис.

Всі користувачі (в тому числі і тимчасові) можуть переглядати інформацію про всі дисципліни. Для цього, з внутрішнього накопичувача, який відповідає користувачу, який переглядає інформацію, надсилається у процес інформація про цього користувача. Також надсилається час перегляду цієї інформації (з зовнішньої сутності User), а також інформація про навчальні дисципліни системи з внутрішнього накопичувача Subject info, і всі переглянуті до цього. Результатом процесу є структура наступного вигляду (таку структуру має внутрішній накопичувач Reviewed pages): user\_login, sbjct\_id, time.

Викладач може видалити дисципліну. Для цього, в процес видалення дисципліни Del subject передається інформація про викладача з внутрішнього накопичувача Teacher, інформація про дисципліну з Subject info, інформація про оцінки з Mark, інформація про роботи з Work. Також у процесі бере участь адміністратор, який перевіряє доцільність видалення дисципліни за допомогою інформації з деканату. Результатом є зміна статусів дисципліни, а також робіт, оцінок, що відносяться до видаленої дисципліни, з поточного значення на значення dropped.

# ВИСНОВКИ

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».

# ПЕРЕЛІК ПОСИЛАНЬ

1. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных: Учебник для высших учебных заведений/Под ред. проф. А.Д. Хомоненко. – СПб.: КОРОНА принт, 2002. – 672с.

2. В.В. Корнеев, А.Ф. Гареев, С.В. Васютин, В.В. Райх Базы данных. Интеллектуальная обработка информации. – М.: Нолидж, 2001.- 496с.

3. Хансен Г., Хансен Д. Базы данных. Разработка и управление. – М.: Бином, 2000. – 704 с.

4. Тимчасове положення про організацію освітнього процесу в НТУУ «КПІ» [Текст] / Уклад.: В. П. Головенкін (розд.: 1-8, 10, 12), С. В. Мельниченко (розд.: 9, 11); за заг. ред. Ю.І. Якименка. – К.: НТУУ «КПІ», 2015. – 102 с.

# Додаток А

Діаграма DFD нульового та першого рівнів процесу «Здача заліку»



Рисунок А.1. Діаграма DFD0 (DFD нульового рівня) процесу «Здача заліку»



Рисунок А.2. Діаграма DFD1 (DFD першого рівня) процесу «Здача заліку»

# Додаток Б

Діаграми DFD другого рівня процесу «Здача заліку»



Рисунок Б.1. Діаграма DFD2 (DFD другого рівня) процесу «Здача заліку», процес 1.2



Рисунок Б.2. Діаграма DFD2 (DFD другого рівня) процесу «Здача заліку», процес 1.1