

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерних наук

ДУБЧАК Руслан Іванович

**Web-орієнтована програмна система
нарахування стипендії/ Web-oriented software
system of stipend charges**

напрямок підготовки: 6.050103 - Програмна інженерія
фахове спрямування - Програмне забезпечення систем

Бакалаврська дипломна робота

Виконав студент групи ПЗС-41
Р. І. Дубчак

Науковий керівник:
викладач КРЕПИЧ С.Я.

Бакалаврську дипломну роботу
допущено до захисту:

"__" _____ 20__ р.

Завідувач кафедри

_____ **А. В. Пукас**

ТЕРНОПІЛЬ - 2016

РЕЗЮМЕ

Дипломна робота містить 95 сторінок, 12 таблиць, 43 рисунки, список використаних джерел із 21 найменування та 4 додатки.

Метою дипломної роботи є розробка веб-орієнтованої системи нарахування стипендії.

Об'єкт дослідження – процес нарахування стипендії.

Предмет дослідження – методи та засоби для розробки веб-орієнтованої системи нарахування стипендії.

Методи розробки базуються на мові програмування PHP та СУБД MySQL

Результатом є готова веб-орієнтована система нарахування стипендії.

Ключові слова: ОКР, стипендія, рейтинг, пільгова категорія.

SUMMARY

Thesis contains 95 pages, 12 tables, 43 figures, list of sources with 21 titles and 4 applications.

The aim of the thesis is the development a web-oriented software system of stipend charges.

The object of research is a means the process of stipend charges.

The subject of research is development of web-based software to the system of stipend charges.

Technologies used to develop software are programming language PHP and RDBMS MySQL.

The result is ready web-oriented software system of stipend charges.

Keywords: education level, scholarship, rating, preferential category.

ЗМІСТ

ВСТУП

РОЗДІЛ I АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Коротка характеристика об'єкту управління 0

1.2 Опис предметної області

1.3 Огляд і аналіз існуючих аналогів 0

1.4 Специфікація вимог до модуля (системи)

Висновки до розділу I 1

РОЗДІЛ II ПРОЕКТУВАННЯ

2.1 Розробка архітектури програмної системи 4

2.2 Проектування структури бази даних

Висновки до розділу II 6

РОЗДІЛ III ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Програмна реалізація проекту 2

3.2 Програмна реалізація бази даних

Висновки до розділу III 3

РОЗДІЛ IV ТЕСТУВАННЯ ТА ДОСЛІДНА ЕКСПЛУАТАЦІЯ

4.1 Тестування 3

4.2 Розгортання програмного продукту

4.3 Інструкція користувача 8

Висновки до розділу IV

ВИСНОВКИ 1

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК А ВИХІДНИЙ КОД РОЗРОБКИ ВЕБ-ОРІЄНТОВАНОЇ 2

СИСТЕМИ НАРАХУВАННЯ СТИПЕНДІЇ

ДОДАТОК Б РЕЗУЛЬТАТ ФІЗИЧНОГО ПРОЕКТУВАННЯ БД 2

ДОДАТОК В РЕЗУЛЬТАТИ ТЕСТУВАННЯ

ДОДАТОК Г ФУНКЦІОНАЛЬНІ ТЕСТОВІ ВИПАДКИ 9

1

2

2

7

8

4

5

6

7

8

7

3

ВСТУП

Актуальність обраної теми, для написання дипломної роботи зумовлена тим, що процес нарахування стипендії та пов'язані з ним маніпуляції посідають значне місце у функціонуванні будь-якого ВНЗ. Уже існуючі програмні продукти-аналоги дозволяють значно полегшити працівнику роботу із даними стипендіатів. Проте великим мінусом таких програм є те, що для нормального їх функціонування (в багатьох випадках) потрібно встановлювати додаткове ПЗ. Також до мінусів можна віднести те, що працівник обмежений в роботі із програмою лише своїм робочим місцем. Крім того, на сьогоднішній день не існує програмного рішення, яке б в комплексі мало змогу працювати із оцінками студентів, обраховувати їх середній бал та, відповідно пільгової категорії (при її наявності) і отриманим даним, призначати стипендію та коректно обраховувати її розмір.

Під час заліково-екзаменаційної сесії у працівника ВНЗ постає задача призначення стипендії за допомогою ряду рутинних дій. Задля правильного призначення стипендії витрачається дуже багато часу аби обрахувати середній бал, звірити наявність пільгової категорії та безпосередньо нарахування відповідної суми грошової виплати студентам.

Тому розробка веб-орієнтованої системи, яка б дозволила спростити роботу працівника ВНЗ, без прив'язки до робочого місця з встановленим ПЗ є актуальною задачею.

Метою роботи є аналіз та детальне дослідження уже існуючих програмних продуктів для роботи із стипендіальним фондом і розробка власної веб-орієнтованої програмної системи, на основі проаналізованих програмних рішень, яка б дозволила ряд суміжних операцій по нарахуванні стипендії виконувати в комплексі.

Отже, спираючись на мету, можна виділи такі основні задачі, які ставляться до розроблюваного продукту:

1. Проаналізувати програмні системи по нарахуванню стипендії, які існують на сьогодні.
2. На основі опрацьованого матеріалу розробити функціональні та нефункціональні вимоги до розроблюваної програмної системи.
3. Спроекувати базу даних програмної системи.
4. Обрати методи та засоби реалізації для втілення в життя поставленого завдання.
5. Розробити веб-орієнтовану систему нарахування стипендії згідно з виконаними задачами, які описані в пунктах 1-4.

Об'єкт дослідження – процес нарахування стипендії.

Предмет досліджень – застосування сучасних методів та технологій проектування для створення веб-орієнтованого додатку системи нарахування стипендії.

Під час створення системи буде використано наступні технології, необхідні для розробки інтернет-сервісів:

- HTML – мова розмітки гіпертекстових сторінок;
- PHP – скриптова мова програмування, для розробки веб-додатків;
- JavaScript – об'єктно-орієнтована, динамічна мова програмування, для розробки веб-застосунків, яка здебільшого використовується як частина браузера;
- SQL – мова, яка дає змогу взаємодіяти із даними, які зберігаються у таблицях БД та виконувати різного роду маніпуляції із ними.

Практична цінність розроблюваного програмного продукту полягає в наданні працівнику ВНЗ швидко організувати процес нарахування стипендії засобами самого програмного продукту, виключаючи при цьому ряд рутинних дій, і наданні можливості доступу до системи із будь-якого місця з покриттям мережі Інтернет.

РОЗДІЛ І

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Коротка характеристика об'єкту управління

Вищий навчальний заклад (далі ВНЗ) — освітній заклад, за допомогою якого отримують вищу професійну освіту та здійснюють навчальну діяльність. Існують приватні та державні заклади. ВНЗ може мати представництва і філіали в інших населених пунктах, вони різняться рівнем акредитації (один з чотирьох). Кожен навчальний заклад має свій статут і являється автономним суб'єктом правових відносин.

На сьогоднішній день фактично кожен навчальний заклад потребує автоматизованої системи для нарахування стипендії.

Управління ВНЗ здійснюється за допомогою багатьох відділів та підрозділів, які певним чином взаємодіють між собою. В залежності від типу закладу та його внутрішнього устрою ці підрозділи можуть носити різну назву, проте їх функціональні зобов'язання залишаються незмінними. Далі розглянемо основні структурні підрозділи, які мають зв'язок із предметною областю даної роботи.

На рисунку 1.1 зображено загальну схему структурної організації управління, яка забезпечує контроль навчального процесу.

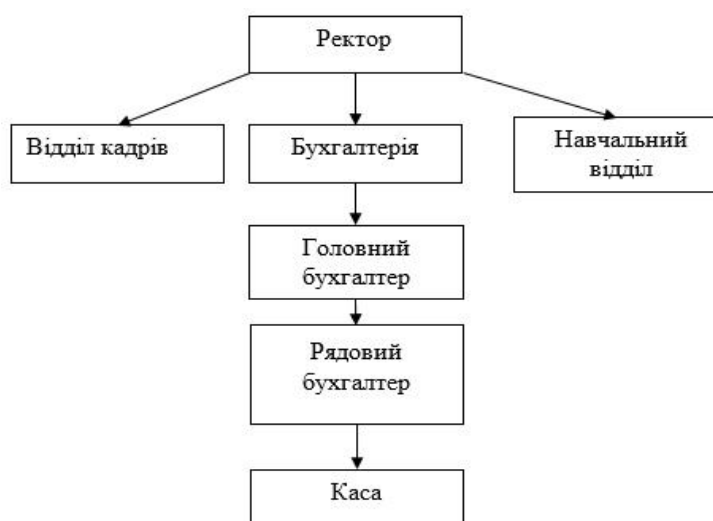


Рисунок 1.1 – Схема організаційної структури управління ВНЗ

Коротко опишемо основні структурні елементи схеми, які представлені на рисунку 1.1:

- ректор – очолює правління навчального закладу і його підрозділів;
- відділ кадрів – відділ, основними функціями якого є управління персоналом університету;
- навчальний відділ – структурний підрозділ, основними напрямками роботи якого є координація наукової діяльності;
- бухгалтерія – структурний підрозділ, що призначений для ведення бухгалтерського обліку і нагляду за фінансовою діяльністю установи;
- головний бухгалтер – керівник відділу «бухгалтерія», який забезпечує коректне ведення бухгалтерського обліку установи;
- бухгалтер – спеціаліст, який займається фінансовим обліком установи, відповідно до чинного законодавства;
- каса – підрозділ, який виконує різного роду операції з готівкою.

1.2 Опис предметної області

Для належної організації роботи веб-орієнтованої системи нарахування стипендії необхідне виконання таких основних бізнес-процесів (див. рис. 1.2):

- процес опрацювання даних студента;
- процес нарахування стипендії;
- процес формування звітності.



Рисунок 1.2 – Діаграма бізнес-процесів розроблюваного програмного продукту

Розглянемо детальніше кожен з вище представлених бізнес-процесів. На рисунку 1.3 бачимо діаграму функцій процесу опрацювання даних студента.

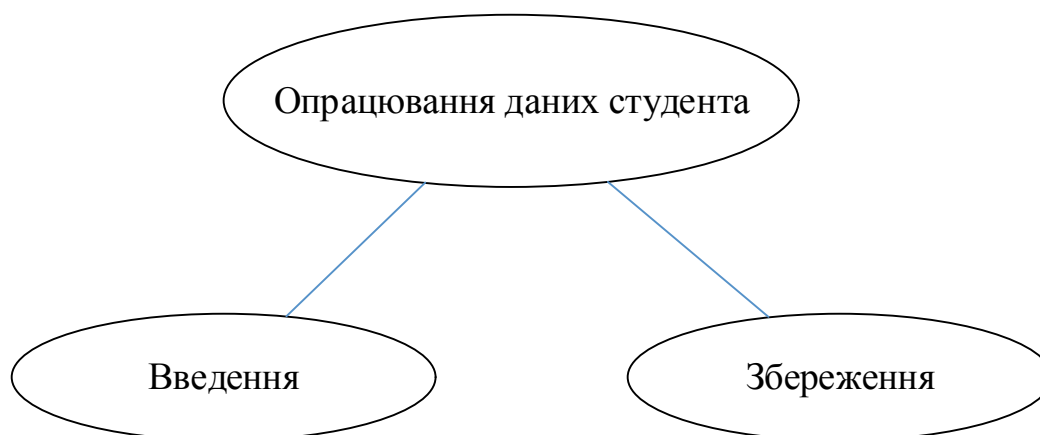


Рисунок 1.3 – Діаграма функцій процесу «Опрацювання даних студента»

Першим етапом роботи з системою є опрацювання даних студента. Цей етап складається з:

- введення даних студента (користувач заповняє відповідні поля персональними даними студента);
- збереження (дані зберігаються у БД).

Характеристику бізнес-процесу «Опрацювання даних студента» наведено в таблиці 1.1.

Таблиця 1.1

Характеристика процесу опрацювання даних студента

Назва характеристики	Значення характеристики
Ім'я бізнес-процесу	Опрацювання даних студента
Основні учасники	Працівник
Вхідна подія	Введення даних студента
Вхідні документи	Паспорт, інд. код, атестат, відмітки про успішність (атестат, відомість і т. п.)
Вихідна подія	Збережений у БД кортеж із даними студента
Вихідні документи	-
Клієнт бізнес-процесу	Нарахована стипендія студенту

На рисунку 1.4 бачимо діаграму функцій процесу «Нарахована стипендія студенту».



Рисунок 1.4 – Діаграма функцій процесу «нарахована стипендія студенту»

Цей процес складається з трьох основних пунктів, а саме:

- аналіз успішності (на основі відомості визначається можливість студента отримувати стипендію);
- вибір пільгової категорії (можливий, лише при її наявності);
- нарахування (нараховується стипендія студенту).

Характеристику бізнес-процесу «Нарахована стипендія студенту» наведено в таблиці 1.2.

Таблиця 1.2

Характеристика бізнес-процесу «Нарахована стипендія студенту»

Назва характеристики	Значення характеристики
Ім'я бізнес-процесу	Нарахована стипендія студенту
Основні учасники	Працівник
Вхідна подія	Аналіз успішності та вибір пільгової категорії (при її наявності)
Вхідні документи	відмітки про успішність (з атестата, відомості і т. п.), посвідчення пільговика
Вихідна подія	Нарахована стипендія студенту
Вихідні документи	-
Клієнт бізнес-процесу	Формування звітності

На рисунку 1.5 бачимо діаграму функцій процесу формування звітності.

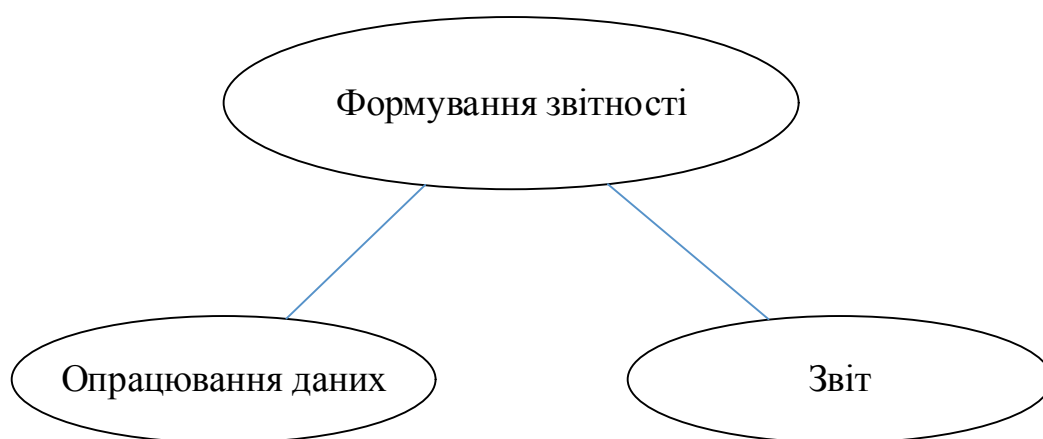


Рисунок 1.5 – Діаграма функцій процесу «Формування звітності»

Цей процес складається таких пунктів:

- опрацювання даних (аналіз даних про нараховану стипендію);
- звіт (формування звітів, платіжних відомостей і т. п.).

Таблиця 1.3

Характеристика процесу «Формування звітності»

Назва характеристики	Значення характеристики
Ім'я бізнес-процесу	Формування звітності
Основні учасники	Працівник
Вхідна подія	Запит на формування звіту
Вхідні документи	-
Вихідна подія	Сформований звіт
Вихідні документи	Платіжна відомість і т. п.
Клієнт бізнес-процесу	-

1.3 Огляд і аналіз існуючих аналогів

На етапі проектування програмного продукту доцільно розглянути існуючі на даний час програми, які безпосередньо або опосередковано виконують функції призначення та нарахування стипендії. Розглянемо наступні програмні продукти: модуль «Обрахування стипендії» для типової конфігурації «1С: Зарплата і кадри бюджетної установи 8» на платформі 1С 8.2 від компанії Infosuite, програмний модуль «БИТ. Расчёт стипендий» для підприємства 1С

компанії «Первый БИТ», модуль «Призначення стипендії» для «1С:Підприємства».

Модуль «Обрахування стипендії» компанії Infosuite по своїй суті являє собою прикладне рішення, призначене для обліку студентів і обрахунку стипендії, яке забезпечує:

- облік студентів по факультетах і групам;
- нарахування стипендій різного типу;
- нарахування утримань;
- розрахунок і виплату стипендії з врахуванням утримань;
- формування звітності по стипендії.

Функціональні можливості:

- зберігання інформації про кожного студента, в розрізі груп, підрозділів (факультетів) і курсів з використанням типових довідників «Працівники» та «Фізичні особи»;
- зберігання списків груп і курсів в спеціальному довіднику «Факультети і групи»;
- можливість внесення ставок стипендії за різні періоди;
- реєстрація кадрових документів (прийом студентів в групи, переміщення студентів із однієї групи в іншу, відрахування (випуск) студентів);
- підбір студентів по параметрах у всіх документах;
- реєстрація даних про розмір і бухгалтерський облік різного типу стипендій
- реєстрація (додавання, зміна, видалення) утримань;
- використання типових механізмів обрахунку заробітної плати для обрахування стипендії;
- використання механізму виправлень для розрахункових документів;
- виплата стипендій, як через касу, так і на банківські картки (рахунки);
- виплата матеріальної допомоги, як оподаткованої, так і неоподаткованої;
- депонування невиплачених сум стипендій;
- реєстрація і проведення утримань за виконавчими листами;

- формування різного типу звітів по стипендіях.

Інтерфейс програми зображено на рисунку 1.6

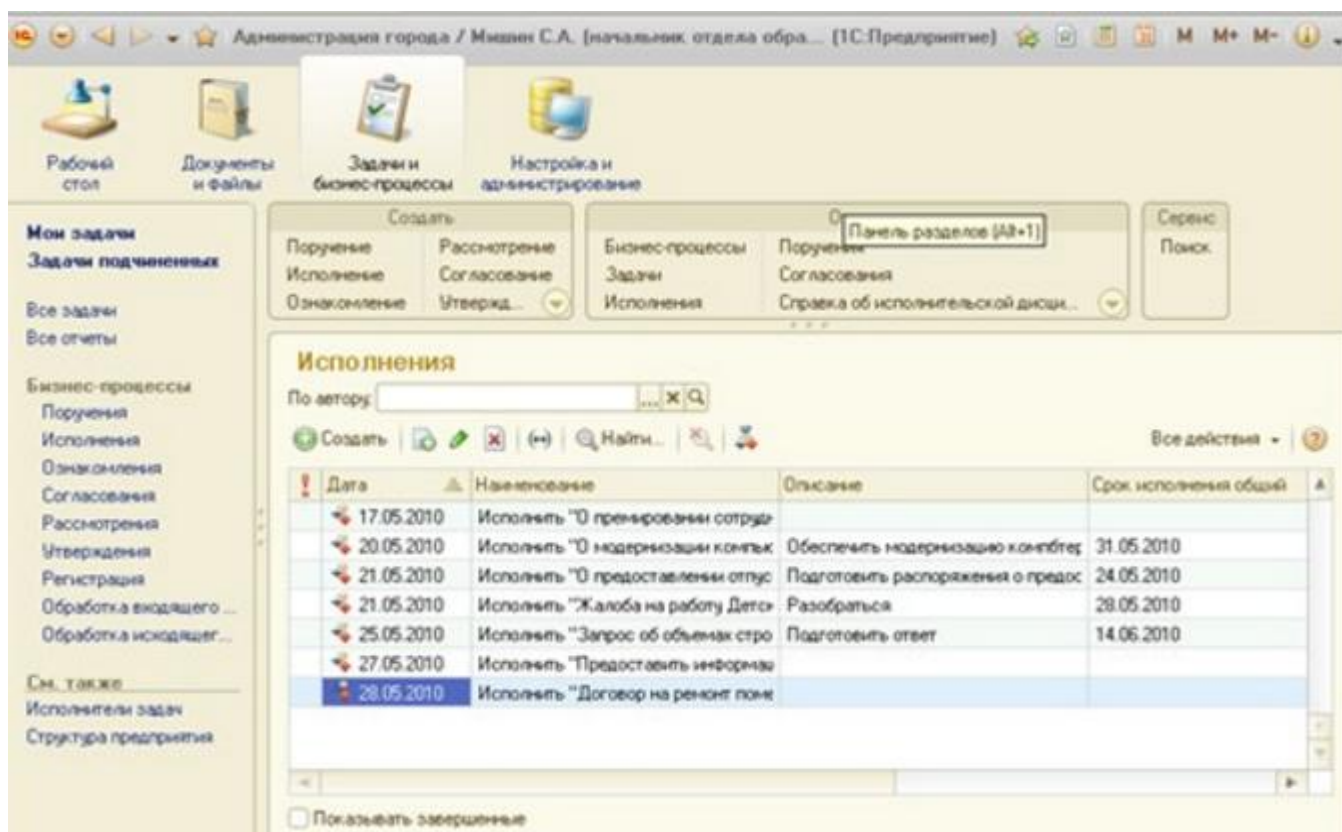


Рисунок 1.6 – Інтерфейс програмного модуля компанії Infosite

«БИТ. Расчет стипендий 8» - програмна система для автоматизації розрахунку стипендій, кадрового обліку студентів, аналізу планових і фактичних нарахувань з можливістю деталізації в розрізі джерел фінансування. Програма вирішує завдання:

- кадрового обліку студентів;
- розрахунку стипендій та інших надбавок;
- виплата і депонування стипендій;
- розрахунок ПДФО і надсилання даних до податкової інспекції;
- надання всієї необхідної звітності по стипендії;
- формування бухгалтерської звітності.

Функціональні можливості:

- автоматизація зарахування, переведення з курсу на курс і відрахування студентів;

- реєстрація даних про розмір і бухгалтерський облік стипендій, реєстрація довільної кількості надбавок і доплат, реєстрація постійних і разових утримань;
- розрахунок ПДФО і надсилання даних до податкової;
- формування платіжних відомостей в банк і касу, депонування;
- відображення в бухгалтерському обліку операцій по нарахуванню і утриманню;
- формування звітності по нарахуванню, розрахункових лисках, звітах по виплаті стипендій та нарахованих податках, книг депонентів, розрахунково – платіжних відомостях та ін.;
- аналіз нарахувань і виплат за джерелами фінансування;
- деталізований звіт по студенту: кадрова інформація, нарахування та утримання за певний період.

Вигляд головного вікна програми зображено на рисунку 1.7

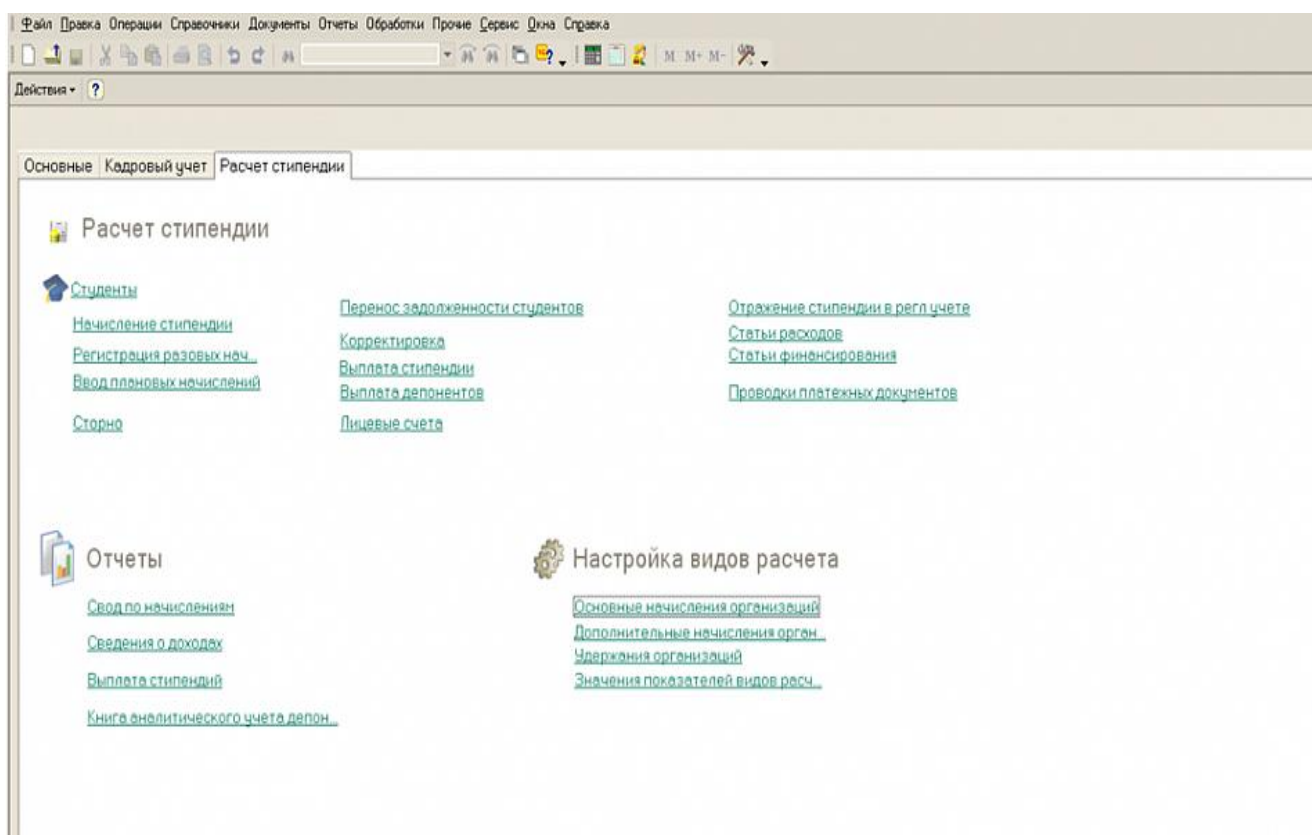


Рисунок 1.7 – Головне вікно програмного модуля «БИТ.Расчет стипендий 8»

Як видно з рисунку 1.7 за допомогою програмного модуля «БИТ.Расчет стипендий 8» можна виконувати такі операції:

1. Розрахунок стипендій:

- нарахування стипендії;
- реєстрація разових нарахувань;
- вивід планових нарахувань;
- перенесення заборгованості студентів;
- корегування;
- виплата стипендії;
- виплата депонентів;
- операції з особовими рахунками;
- відображення стипендії в регулярних звітах;
- інформація про витрати;
- інформація про нарахування;
- проведення платіжних документів.

2. Формування звітів:

- звіт про нарахування;
- звіт про дохід;
- звіт про виплату стипендій;
- Книга аналітичного обліку депонентів.

3. Налаштування виду розрахунку:

- основні нарахування організацій;
- додаткові нарахування організацій;
- утримання організацій;
- значення показників видів розрахунку.

Інформація про кожного студента, список студентів в розрізі груп, підрозділів (факультетів) і курсів зображено на рисунку 1.8 та рисунку 1.9

Студенты: Воронов Олег Петрович

Наименование: **Воронов Олег Петрович** [История Ф.И.О.](#)

Код: 000000009 Категория студента: Хорошист

Основные Адреса Кадровые документы Начисления Дополнительно

Дата рождения: 17.09.1992

Пол: Мужской

Гражданство: РОССИЯ

Является: Resident с 01.01.2012 [Изменить статус](#)

Удостоверение личности: Паспорт иностранного гражданина, серия: 12 36, № 123654, выдан: 07 дект

ИНН: 772970013388

Банк: Банк Москвы

Счет: 36305690899987777

Выплачивать через кассу

Комментарий:

OK Записать Закрыть

Рисунок 1.8 – Відомості про конкретного студента

Як видно з рисунку 1.8 в зображеному вікні програми можна переглядати та редагувати основні відомості про студента, а також відомості про нарахування стипендії. Також доступним є перегляд кадрової документації.

По группам	Код	ФИО	Подразделение	Курс	Комментарий	Дата за...	Дата от...	Категория	Стипендия	Банк
Студенты	000000	Арипов Антон Степа...	Кафедра экономики	1 курс		01.09.20...		Отличник		Банк Москвы
	000000	Веденцова Елена Вик...				01.09.20...	01.01.2...			Банк Москвы
	000000	Викторов Денис Вале...	Кафедра философии	2 курс		01.09.20...				
	000000	Воронов Олег Петров...	Факультет электро...	1 курс		01.09.20...		Хорошист		Банк Москвы
	000000	Губова Анастасия				01.09.20...	01.12.2...			Банк Москвы
	000000	Давыдов Игорь Иван...	Факультет электро...	1 курс		01.09.20...				
	000000	Иванова Ольга Серге...	Кафедра права	1 курс		01.09.20...		Отличник		
	000000	Каложный Богдан Ва...	Факультет электро...	1 курс		01.09.20...		Хорошист		Банк Москвы
	000000	Красичина Валентина	Кафедра права	1 курс		01.09.20...				
	000000	Олейник Юлия Влади...	Кафедра права	1 курс		01.09.20...				
	000000	Романов Илья Васил...	Кафедра права	1 курс		01.09.20...				
	000000	рыкова галина	Факультет электро...	2 курс		01.09.20...				Банк Москвы
	000000	Семенов Кирилл Евге...	Кафедра физическо...	1 курс		01.09.20...		Отличник		
	000000	Сергеева Анна Ивано...				01.09.20...				Банк Москвы
	000000	Смирнов Максим Ива...	Кафедра права	Магистрату...		01.01.20...			1 500,00	
	000000	Сорокин Владислав Е...								

Рисунок 1.9 – Списки студентів

Як видно з рисунку 1.9 в зображеному вікні програми відкривається можливість перегляду списку студентів та основної інформації відповідно до сформованих груп (випускники, відраховані, працівники академії, студенти).

На рисунку 1.10 зображено вікно, яке демонструє автоматизацію щомісячного нарахування стипендії, обчислення нарахувань, утримань, ПДФО. Документ можна заповнити як вручну, так і автоматично.

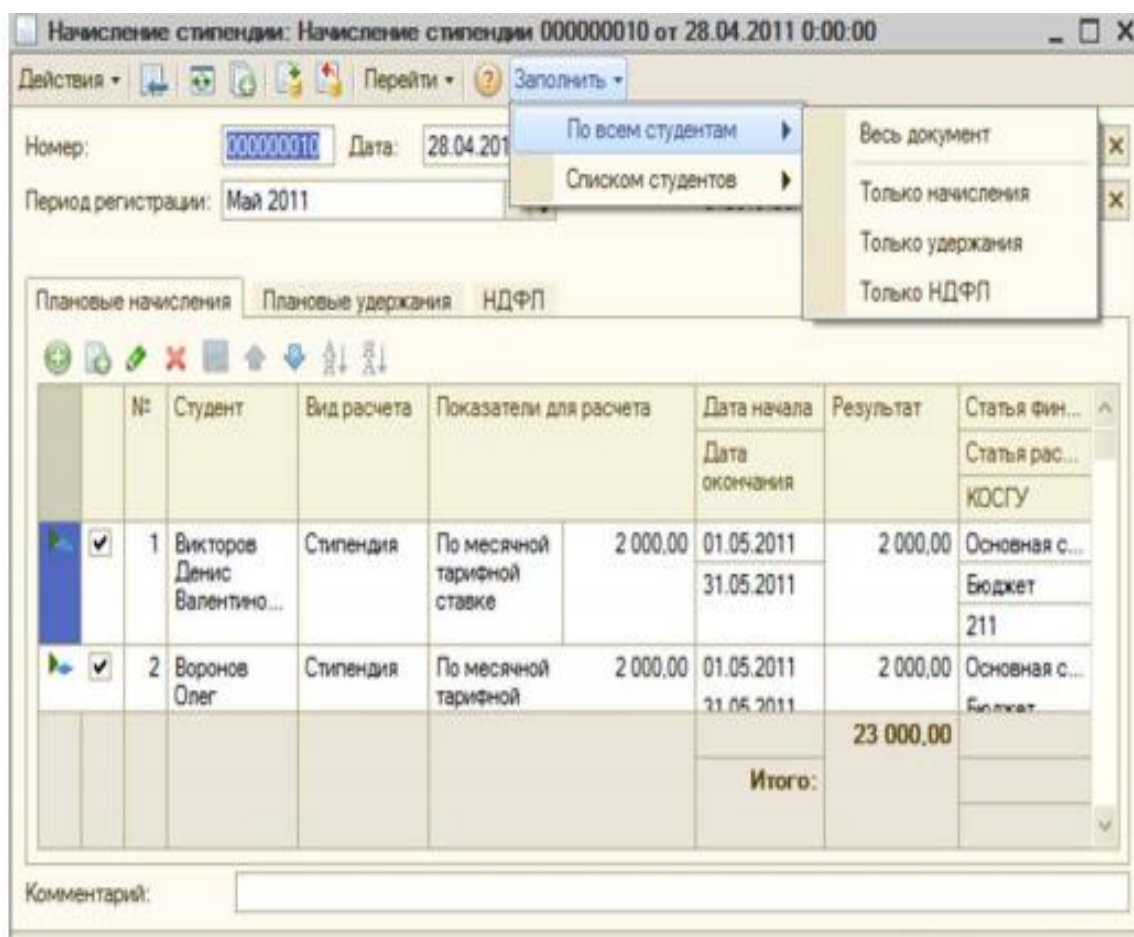


Рисунок 1.10 – Вікно нарахування стипендії

Як видно з рисунку 1.10 в зображеному вікні програми відкривається можливість автоматичного заповнення нарахувань по стипендії, за певний період.

Результатом проведеної операції, представленої на рисунку 1.10, є сформована платіжна відомість, яку показано на рисунку 1.12

Выплата стипендии: Выплата стипендии 000000021 от 30.11.2010 12:00:00

Действия: Перейти Заполнить

Номер: 000000021 от: 3 Организация: МАУ
 Период регистрации: Ноябрь 2010 Ответственный: Глебова А.И.
 Статья финансирования: КОСГУ: Банк: Банк Москвы
 Вид выплаты: Ведомость в банк Дата выплаты: 30.11.2010
 Округлять (при заполнении)

	№	Студент	Лицевой счет	Сумма	Статья финансирования КОСГУ
	1	Воронов Олег Петрович	369856989899...	3 000,00	Основная статья финансирования 290
	2	Веденева Елена ...	369852147855...	2 000,00	Основная статья финансирования 290
	3	рыкова галина	236598745412...	2 970,00	Внебюджетная статья 290
	4	рыкова галина	236598745412...	5 220,00	Основная статья финансирования
		Итого:		27 190,00	

Комментарий: _____

Задепонированные Суммы | Печать | OK | Записать | Закрыть

Рисунок 1.11 – Вікно виплати стипендії

Як видно з рисунку 1.11 в зображеному вікні програми відкривається можливість заповнення платіжної відомості та супровідних атрибутів. Після обробки відомість можна відправити на друк. Вигляд сформованої платіжної відомості зображено на рисунку 1.12

В кассу для оплаты в срок
с _____ по _____
в сумме Шестнадцать тысяч рублей 00 копеек (

Проректор _____
Гл. бухгалтер _____

ПЛАТЕЖНАЯ ВЕДОМОСТЬ № 000000005
за Сентябрь 2010 г.

Форма по ОКУД _____
Дата _____
по ОКПО _____
по ОКЕИ _____

Исчисление: МАУ
Структурное подразделение: _____
Единица измерения: руб

Дебет счета: _____ Кредит счета: _____ Расходный кассовый ордер № _____ Дата: _____

№ п/п	Фамилия, имя, отчество	Табельный номер	Сумма	Расход
1	2	3	4	
1	Семенов Кирилл Евгеньевич	0000000010	2 000,00	
1	Олейник Юлия Владимировна	0000000012	2 000,00	
1	Викторов Денис Валентинович	0000000014	2 000,00	
1	Калюжный Богдан Вадимович	0000000016	2 000,00	
1	Давыдов Игорь Иванович	0000000017	2 000,00	
1	Романов Илья Васильевич	0000000018	2 000,00	
1	Иванова Ольга Сергеевна	0000000019	2 000,00	
1	Крапивина Валентина Романовна	0000000020	2 000,00	
	Итого:		16 000,00	

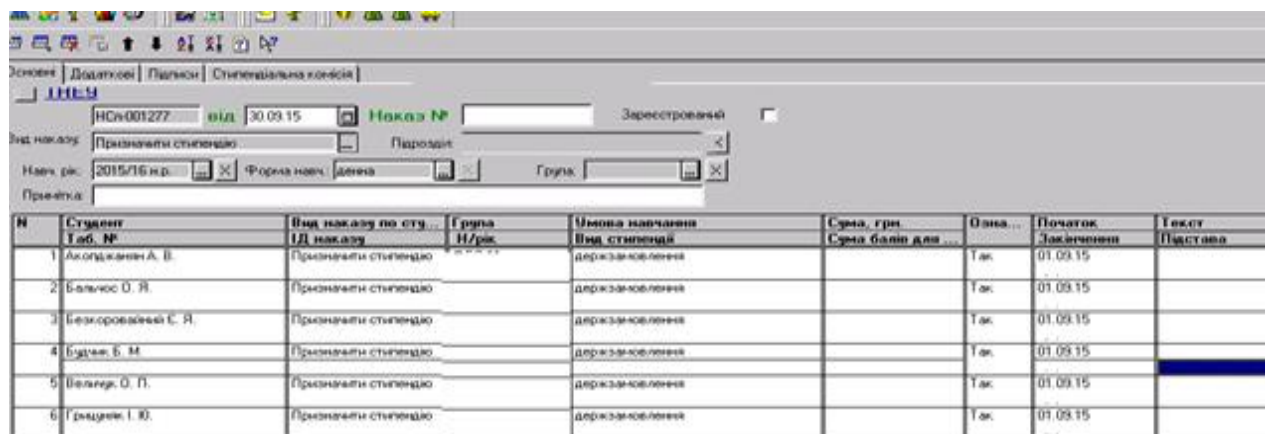
Рисунок 1.12 – Платіжна відомість

Модуль «Призначення стипендії ТНЕУ» для підприємства 1С застосовується для автоматизації процесу нарахування стипендії. Включає в себе можливість ведення журналу успішності, за підсумками сесій та списків пільговиків, відповідно до факультету.

Функціональні можливості:

- зберігання основної інформації про студента, його успішності;
- ведення реєстру пільговиків;
- формування наказів призначення стипендії;
- вибір типу стипендії, відповідно досягнень, освітньо-кваліфікаційного рівня та пільгової категорії (за наявності).

Наказ на призначення стипендії зображено на Рисунку 1.13



№	Студент Таб. №	Вид наказу по ст... ID наказу	Група Н/урк	Типова назва Вид стипендії	Сума, грн. Сума балів для...	Озна...	Початок Закінчення	Текст Підстава
1	Акопджанян А. В.	Призначити стипендію		держзамовлення		Так	01.09.15	
2	Бальчос О. Я.	Призначити стипендію		держзамовлення		Так	01.09.15	
3	Безжуровський С. Я.	Призначити стипендію		держзамовлення		Так	01.09.15	
4	Будяк Б. М.	Призначити стипендію		держзамовлення		Так	01.09.15	
5	Веллер О. П.	Призначити стипендію		держзамовлення		Так	01.09.15	
6	Григорук І. Ю.	Призначити стипендію		держзамовлення		Так	01.09.15	

Рисунок 1.13 – Вікно наказу на призначення стипендії

Як видно з рисунку 1.13, наказ на призначення стипендії формується кожним факультетом університету окремо, при чому формування списку студентів-стипендіатів відбувається вручну шляхом почергового додавання груп. При чому вид стипендії, яку призначатимуть студенту потрібно обирати для кожного окремо, відповідно до середнього балу за сесію та пільгової категорії, якщо вона є.

Сформована відомість успішності групи за семестр зображена на рисунку 1.14

Оновити		Настройка		2014/15 н.р.											Середні
№ з/п	№ заліковки	Прізвище, ім'я, по батькові студента	2 семестр											у 5-бальній шкалі	
			Безпека життєдіяльності	Вища математика	Іноземна мова	Інформатика	Історія України	Історія української культури	Макроекономіка	Основи наукових досліджень	Психологія і педагогіка	Українська мова (за професійним спрямуванням)	Фізичне виховання		
1	2	3	Екзамен	Екзамен	Залік	Залік	Екзамен	Екзамен	Екзамен	Залік	Залік	Екзамен	Залік	15	
1	01119/14	ГНАТІВ Ольга Віталіївна	5	5	5	5	5	5	5	5	5	5	5	5.00	
2	01132/14	РУДА Олена Ярославівна	4	5	4	4	5	4	5	4	4	5	4	4.36	
3	01134/14	СИДОРОВИЧ Юлія Іванівна	4	5	5	4	4	4	5	4	4	5	4	4.36	
4	01125/14	МАНЬКО Михайло Віталійович	4	4	4	4	5	4	4	5	5	4	4	4.27	
5	01123/14	ЗУБИК Ірина Леонідівна	5	4	4	4	4	4	4	4	4	4	4	4.18	
6	01126/14	МЕЛЬНИК Ірина Любомирівна	4	4	4	4	4	4	5	5	4	4	4	4.18	
7	01130/14	ПОЛЯНСЬКА Марія Іванівна	5	3	5	4	4	4	4	4	4	4	4	4.09	
8	01136/14	СУХІНСЬКА Лілія Павлівна	4	4	5	4	4	4	4	4	4	4	4	4.09	
9	01137/14	ТКАЧУК Валерій Володимирович	4	4	4	4	5	4	4	4	3	4	5	4.09	
10	01115/14	БАРИЛЮК Тарас Михайлович	4	4	5	4	4	4	4	4	4	4	3	4.00	
11	01121/14	ДРАБИК Софія Петрівна	5	3	4	4	4	4	4	3	4	4	5	4.00	
12	01138/14	ФЕРЕНЦ Назарій Юрійович	4	4	5	4	4	4	3	4	4	4	4	4.00	
13	01139/14	ФІЛЬВАРОЧНИЙ Ігор Васильович	4	4	4	4	4	4	4	4	4	4	4	4.00	
14	01140/14	ЯЛОВЕГА Ярослав Володимирович	4	4	4	4	4	3	4	3	4	4	4	3.82	
15	01128/14	ОЛЕКСИН Роман Михайлович	4	3	4	4	4	4	4	3	4	3	4	3.73	
16	01116/14	ВЕНГЕР Андрій Васильович	3	3	3	3	3	3	3	3	3	3	5	3.18	
Складали іспит:			16	16	16	16	16	16	16	16	16	16	16		
Відмінно (А)			4	3	6	1	4	1	4	3	2	3	5		
Добре (В, С)			11	9	9	14	11	13	10	9	12	11	10		
Задовільно (D, E)			1	4	1	1	1	2	2	4	2	2	1		
Незадовільно (F)			-	-	-	-	-	-	-	-	-	-	-		
Повторний курс (F)			-	-	-	-	-	-	-	-	-	-	-		
Неявка поважна			-	-	-	1	-	-	-	-	1	1	-		
Неявка неповажна			-	-	-	-	-	-	-	-	-	-	-		
Абсолютна успішність:			100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0		
Якість навчання:			93.8	75.0	93.8	93.8	93.8	87.5	87.5	75.0	87.5	87.5	93.8		

Рисунок 1.14 – Вікно успішності

Отже, якщо студент набрав нижче «4» балів він не претендує на стипендію, виключенням є лише студенти-пільговики, яким достатньо набрати «3» бали за сесію. Формування успішності студентів за сесію відбувається вручну для кожної групи. Однак у вікні успішності не виокремлюються студенти-пільговики, тому необхідно додатково переглядати реєстр студентів за пільгами. Вибір студентів пільговиків і сформовані списки пільговиків зображено на рисунках 1.15 та 1.16.

Рисунок 1.15 – Вибір студентів пільговиків

5C4

1	2	3	4	5	6	7	8	9	10	11
1	Тернопільський національний економічний університет									
2	Реєстр пільговиків з 01.07.15 по 30.09.15									
3										
4	факультет									
5		Вид пільги Студент	День народження	Вік	Курс	Група	Розмір стипендії	Статус	Початок дії	Закінчення дії
6	Дитина-інвалід					6				
7	1	ВИСОЦЬКИЙ Павло Володимирович	01.06.96	19	3				01.09.13	..
8	2	ГЛАДІЙ Ірина Михайлівна	15.02.97	18	2				01.09.14	..
9	3	ГЛУЩЕНКО Тарас Миколайович	07.03.96	19	3				01.09.15	..
0	4	ГУДИМА Юлія Андріївна	13.04.97	18	2				01.09.14	..
1	5	ПРОКІПЧУК Лілія Олесівна	12.12.97	17	1				01.09.15	..
2	6	ЧАЙКА Ірина Русланівна	03.10.96	18	2				01.09.14	..
3	Дитина-сирота на повному державному забезпеченні					3				
4	1	БУЛКОВСЬКА Ганна Григорівна	13.12.93	21	1				01.09.12	..
5	2	МАТІЙЦІВ Володимир Володимирович	01.01.93	22	1				01.09.13	..
6	3	ПЕНЧУК Марія Зеновівна	27.08.97	18	3				01.09.15	..
7	Інвалід 2 групи					3				
8	1	БАБН Олександра Олександрівна	06.07.94	21	4				01.09.14	..
9	2	КАТЕРИНЮК Христина Миколаївна	18.06.96	19	3				01.01.15	..
0	3	МУДРИК Катерина Андріївна	19.06.94	21	4				01.09.12	..
1	Інвалід 3 групи					4				
2	1	КОСТИШИН Володимир Сергійович	11.08.94	21	3				01.09.15	..
3	2	ЛИСАК Олександр Васильович	08.09.97	18	1				01.09.15	..
4	3	МЕЛЬНИК Володимир Вікторович	27.07.95	20	3				01.09.15	..
5	4	ПЕЛЕХАТ Дана Володимирівна	17.02.93	22	6				01.09.14	..
6	Всього					16				
7										
8										
9										

Рисунок 1.16 – Сформований список пільговиків по факультету

Тип стипендії, яку призначатимуть студенту необхідно також обирати вручну, відповідно до середнього балу за сесію та пільгової категорії. Форму вибору типу стипендії зображено на рисунку 1.17

The screenshot shows a software interface for assigning stipends. At the top, there are fields for 'НСп-001277', 'від 30.09.15', 'Наказ №', and 'Зареєстрований'. Below these are fields for 'Призначити стипендію', 'Навч. рік: 2015/16 н.р.', 'Форма навч.: денна', and 'Група:'. A table lists 15 students with columns for 'Студент Таб. №', 'Вид наказу по сту... ІД наказу', 'Група Н/рік', and 'Точаток Закінчення'. A dropdown menu is open, showing a list of stipend types, with 'Базова стипендія інвалідан' highlighted.

Студент Таб. №	Вид наказу по сту... ІД наказу	Група Н/рік	Точаток Закінчення
1	Акопджанян А. В.	Призначити стипендію	01.09.15
2	Бальчос О. Я.	Призначити стипендію	01.09.15
3	Безкоровайний Є. Я.	Призначити стипендію	01.09.15
4	Будчик Б. М.	Призначити стипендію	01.09.15
5	Величук О. П.	Призначити стипендію	01.09.15
6	Гришуняк І. Ю.	Призначити стипендію	01.09.15
7	Давидюк О. В.	Призначити стипендію	01.09.15
8	Довгань А. В.	Призначити стипендію	01.09.15
9	Дурай П. А.	Призначити стипендію	01.09.15
10	Кондзьолка Р. Я.	Призначити стипендію	01.09.15
11	Остапчук А. В.	Призначити стипендію	01.09.15
12	Рудик Н. М.	Призначити стипендію	01.09.15
13	Швед О. Г.	Призначити стипендію	01.09.15
14	Шикеринєць І. Ю.	Призначити стипендію	01.09.15
15	Яцина Д. Р.	Призначити стипендію	01.09.15

Рисунок 1.17 – Вибір типу стипендії

Проте варто відмітити, що процес визначення можливості отримання стипендії студентом, та процес ідентифікації пільгової категорії(при її наявності) здійснюється вручну. Для кожного освітнього рівня прохідним балом, для отримання стипендії є «4», для отримання підвищеної стипендії «5». Але існують й виключення, при наявності пільгової категорії. Розмір отримуваної стипендії, відповідно до раніше згаданих критеріїв визначається працівником.

Весь цей процес можна спростити, шляхом впровадження певних програмних рішень, за допомогою яких аналіз даних працівником “вручну” - відійшов би на задній план. Цим самим зменшився б вплив людського фактору, мінімізуючи можливість помилки в процесі призначення стипендії.

У таблиці 1.4 представлено порівняльну характеристику програмних продуктів.

Таблиця 1.4

Порівняльна характеристика програмних продуктів

Фірма-розробник	Infosuite	«Первый БИТ»	ІС:Підприємство
Тип продукту	Програмний модуль для підприємства ІС	Програмний модуль для підприємства ІС	Програмний модуль для підприємства ІС
Дата заснування	18-03-2013	Лютий, 2010р	-
Місце знаходження	Росія, Москва	Росія, Москва	-
Версії продукту	Клієнтська ліцензія на 1 робоче місце	БИТ. Расчёт стипендий. Клієнтська ліцензія на 1 робоче місце	1.0
Функціональність	достатня	середня	середня
Інтерфейс користувача	Інтуїтивно-зрозумілий	Простий, одні і ті ж дії можуть бути викликані як за допомогою панелі, так і через робочий стіл	Максимально простий та доступний користувачу
Допомога користувачу	Відсутня	Присутня	Відсутня

1.4 Специфікація вимог до модуля (системи)

Кожна предметна область має свої термінологію. Для забезпечення правильного розуміння користувачем цих понять формується глосарій. Глосарій для обраної предметної області наведено у таблиці 1.5

Таблиця 1.5

Глосарій

Термін	Опис Терміну
1. Основні поняття та критерії предметної області та проекту	
Депонування	передача коштів банківським установам задля їх зберігання на певний термін. Також сюди відноситься передача певної грошової суми (цінних паперів), яку повинен передати боржник у судові органи для подальшої передачі кредитору. [1]
Стипендія	грошова виплата, яку нараховують регулярно (як правило щомісяця) учням і студентам середніх спеціальних та вищих навчальних закладів, а також аспірантам та докторантам, за умови успішного навчання.[2]
Індексація	механізм який спрямований на підвищення доходу громадян, що надає можливість повністю або частково відшкодувати затрачені кошти, на споживчі товари та послуги які підпадають подорожчанню. [3]

Продовження таблиці 1.5

Податкові пільги	бонус, що надається певним верствам населення, який включає в себе можливість сплачувати податок в меншому розмірі або ж не сплачувати його взагалі. [4]
ПДФО (Податок на доходи фіз. осіб)	загальнодержавний податок, який стягується з доходів фізичних осіб (громадян — резидентів) і не резидентів, які отримують доходи з джерел їх походження в Україні. [5]
2. Користувачі системи	
Адміністратор	людина, функціональними зобов'язаннями якої являється виконання усіх видів операцій, доступних у системі.
Головний бухгалтер	керівник відділу бухгалтерії, який забезпечує ведення бухгалтерського обліку в установі. [6]
Бухгалтер	фахівець, функціональними зобов'язаннями якого є відстеження руху коштів, нарахування зарплат та стипендій, ведення бухгалтерської звітності. [7]
3. Вхідні та вихідні документи	
Відомість успішності	містить відмітки про успішність студента, за підсумком сесії.
Наказ на виплату стипендії	містить вказівки на виплату стипендії та всі супроводжуючі атрибути. [8]

Опис внутрішньої роботи системи та її поведінка наведені у специфікації функціональних вимог (таблиця 1.6).

Таблиця 1.6

Специфікація функціональних вимог

Ід. вимоги	Назва (варіанту вимоги)	Атрибути вимог		
		Пріоритет	Складність	Контакт
1	Авторизація користувача	Обов'язкове	середня	Адміністратор, головний бухгалтер, бухгалтер
2	Додавання даних	Обов'язкове	середня	Адміністратор, головний бухгалтер, бухгалтер
3	Редагування даних	Обов'язкове	низька	Адміністратор, головний бухгалтер
4	Опрацювання даних	Обов'язкове	низька	Адміністратор, головний бухгалтер, бухгалтер
5	Формування звітів	Обов'язкове	висока	Адміністратор, головний бухгалтер, бухгалтер

Вимоги до характеру поведінки системи визначаються нефункціональними вимогами. Нефункціональні вимоги до системи наведено у таблиці 1.7.

Таблиця 1.7

Специфікація нефункціональних вимог

Ід. вимоги	Назва вимоги	Атрибути вимог		
		Пріоритет	Складність	Контакт
1.	Застосованість	Рекомендовано	Висока	Адміністратор, головний бухгалтер, бухгалтер
2.	Надійність	Обов'язково	Висока	Адміністратор, головний бухгалтер, бухгалтер
3.	Безпека та захист даних	Обов'язково	Висока	Адміністратор, головний бухгалтер, бухгалтер
4.	Відмовостійкість	Обов'язково	Висока	Адміністратор, головний бухгалтер, бухгалтер

В системі є три типи користувачів: адміністратор, головний бухгалтер та бухгалтер. Для кожного із них система надаватиме різний функціонал. Спільними для усіх видів користувачів будуть функції авторизації, роботи зі студентами, формування звітності та деякі можливості роботи зі стипендією. Відношення між користувачами та варіантами використання зображено на рисунку 1.18

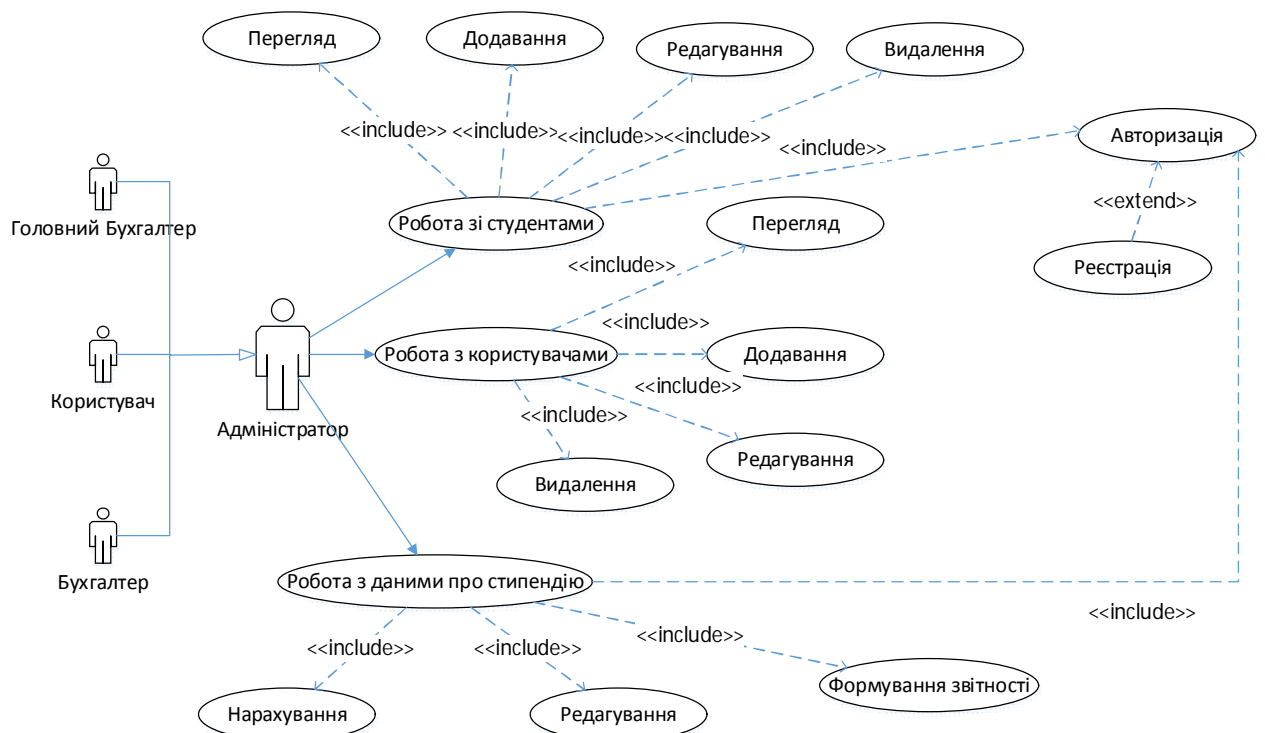


Рисунок 1.18 Діаграма варіантів використання

Розглянемо детальніше кожен із варіантів використання. Для початку роботи з системою потрібно ідентифікуватися. Це можна зробити за допомогою

пункту «Авторизація». Після успішного логування на сайті, система, відповідно до типу облікового запису, відкриє основний функціонал.

У таблиці 1.8 представлено опис варіанту використання «Авторизація».

Таблиця 1.8

Варіант використання «Авторизація»

Контекст використання	Метою функції є ідентифікація користувача та надання йому відповідних можливостей
Дійові особи	Адміністратор, головний бухгалтер, бухгалтер
Передумова	–
Тригер	Перший запуск додатку
Сценарій	1 – Ввід авторизаційних даних 2 – Натискання кнопки підтвердження авторизації
Пост умова	Відкривається основний функціонал системи / повідомлення про помилку

Логічний і концептуальний опис функціональних можливостей системи для сценарію «Авторизація», відображено на ескізі екранної форми, яка представлена на рисунку 1.19

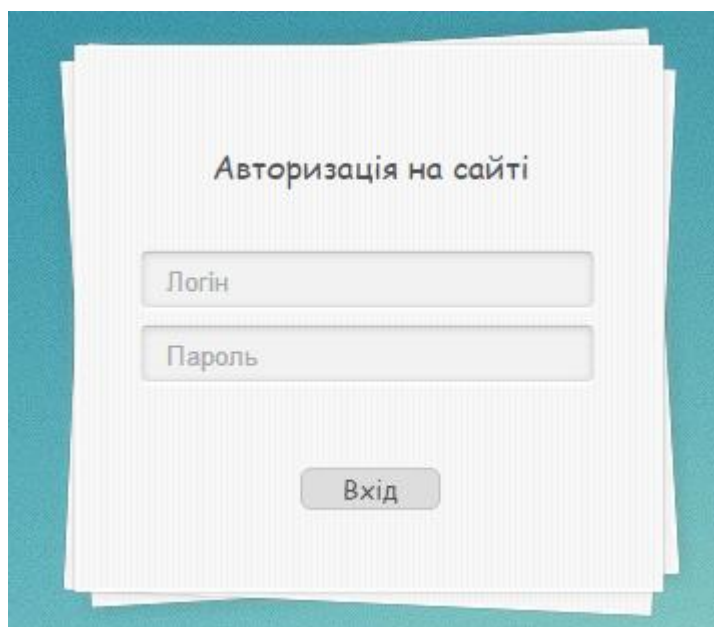


Рисунок 1.19 – Кадр для сценарію «Авторизація»

Далі розглянемо варіант використання пункту «Робота зі студентами -> Перегляд». За допомогою цієї функції перед користувачем системи відкриється можливість перегляду детальної інформації про студентів.

У таблиці 1.9 представлено опис варіанту використання «Робота зі студентами -> Перегляд».

Таблиця 1.9

Варіант використання «Робота зі студентами -> Перегляд»

Контекст використання	Метою функції є формування та відображення даних студента
Дійові особи	Адміністратор, головний бухгалтер, бухгалтер
Передумова	Успішна авторизація
Тригер	Вибір пункту в меню
Сценарій	1 – Вибір пункту «Робота зі студентами» 2 – Вибір пункту «Перегляд»
Пост умова	Відображені дані студентів

Логічний і концептуальний опис функціональних можливостей системи для сценарію «Робота зі студентами -> Перегляд», відображено на ескізі екранної форми, яка представлена на рисунку 1.20

Група	Прізвище	Ім'я	по Батькові	Інд. код	Форма навчання
pzs-41	Андрусяк	Іван	Степанович	135478964	Державник
pzs-41	Бойко	Микола	Ігорович	25646315	Державник
pzs-41	Ігоренко	Петро	Андрійович	457899321	Державник
pzs-41	Ковальчук	Іван	Іванович	1236655484	Державник
pzs-41	Мартиненко	Григорій	Йосипович	235789412	Державник
pzs-41	Назаренко	Назар	Назарович	987654321	Платник
pzs-41	Потоцький	Юрій	Сергійович	458723648	Платник

Рисунок 1.20 – Кадр для сценарію «Робота зі студентами -> Перегляд»

Далі розглянемо варіант використання пункту «Робота зі студентами -> Редагування». За допомогою цієї функції перед користувачем системи відкриється можливість редагування даних, шляхом їх перезапису.

У таблиці 1.10 представлено опис варіанту використання «Робота зі студентами -> Редагування».

Таблиця 1.10

Варіант використання «Робота зі студентами -> Редагування»

Контекст використання	Метою функції є надання можливості редагування уже збережених даних
Дійові особи	Адміністратор, головний бухгалтер, бухгалтер
Передумова	Успішна авторизація
Тригер	Вибір пункту в меню
Сценарій	1 – Вибір потрібного, для внесення змін запису 2 – Введення потрібних даних 3 – Натиснення кнопки підтвердження
Пост умова	Повідомлення про успішне / неуспішне внесення змін

Логічний і концептуальний опис функціональних можливостей системи для сценарію «Робота зі студентами -> Редагування», відображено на ескізі екранної форми, яка представлена на рисунку 1.21

Рисунок 1.21 – Кадр для сценарію «Робота зі студентами -> Редагування»

Висновки до розділу I

В процесі виконання розділу було складено та описано організаційну структуру управління ВНЗ, визначено основні процеси, які реалізують бізнес-логіку обраної предметної області. Також визначено основні функціональні та не функціональні вимоги до веб-орієнтованої системи нарахування стипендії. Розглянуто рівні доступу користувачів до функціоналу, який буде надаватися системою.

Провівши аналіз програмних продуктів ведучих фірм-розробників ПЗ (Infosuite, Первый БИТ та модуль «Призначення стипендії» для «ІС:Підприємства»), можна використати деякі рішення при розробці. А саме: ведення кадрового обліку студентів, розрахунок суми стипендії, можливість ведення звітності. Проте функціонал потребує доопрацювання, оскільки немає єдиної системи, доступ до якої можливий засобами мережі Інтернет, що дозволило б працівнику не прив'язуватись до робочого місця з попередньо-встановленим ПЗ. Також ні один із аналогів немає функції обрахунку суми стипендії відповідно до рейтингу та пільгової категорії без втручання працівника, який виконує це за допомогою ряду рутинних дій, яких можна було б уникнути. Враховуючи ці чинники буде розроблено веб-орієнтований програмний продукт, який вирішить вище-згадані проблеми.

РОЗДІЛ II

ПРОЕКТУВАННЯ

3.1 Розробка архітектури програмної системи

Веб додаток буде використовувати 3 основні компоненти: веб браузер, веб сервер та базу даних. На рисунку 2.1 зображено діаграму компонентів, яка відображає залежності між компонентами програмного забезпечення.

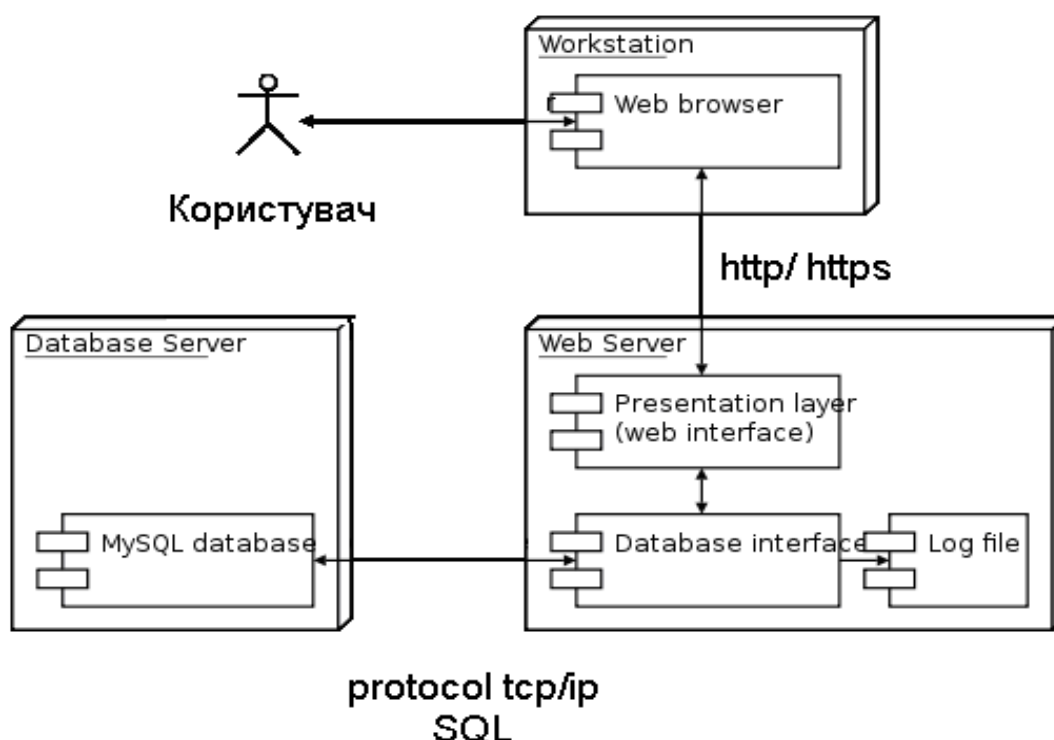


Рисунок 2.1 – Діаграма компонентів розроблюваної програмної системи

Статичне представлення структурної моделі зображають у вигляді діаграми класів. Структурна модель додатку є досить складною. Система матиме велику кількість класів, що забезпечуватимуть можливість її функціонування. Більшість модулів системи відповідатимуть за інтерфейс або стосуватимуться особливостей налагодження роботи. Тому їх відображення на даному етапі є не потрібним оскільки воно зробить діаграму класів занадто громіздкою та важкою для читання і сприйняття. Отже на рисунку 2.2 діаграма відображена лише тими класами, які забезпечуватимуть виконання логіки програми.

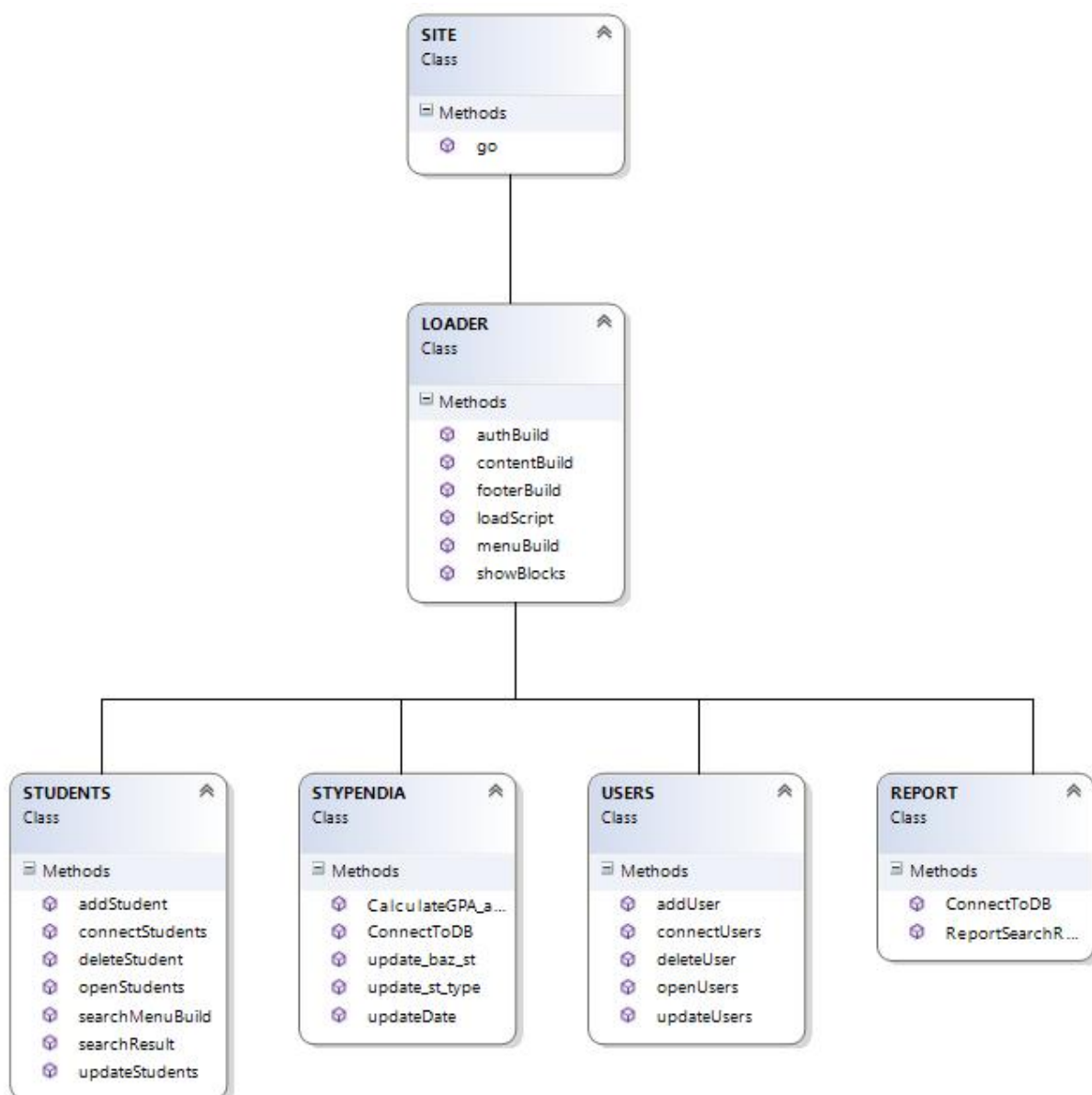


Рисунок 2.22 - Діаграма класів

На рисунку 2.3 зображена діаграма послідовності для варіанту використання «Редагування користувачів», яка відображає взаємодію об'єктів впорядкованих за часом. Можливість маніпулювання із користувачами буде доступна лише адміністратору.

Вихідний код розробки класів наведено у додатку А.

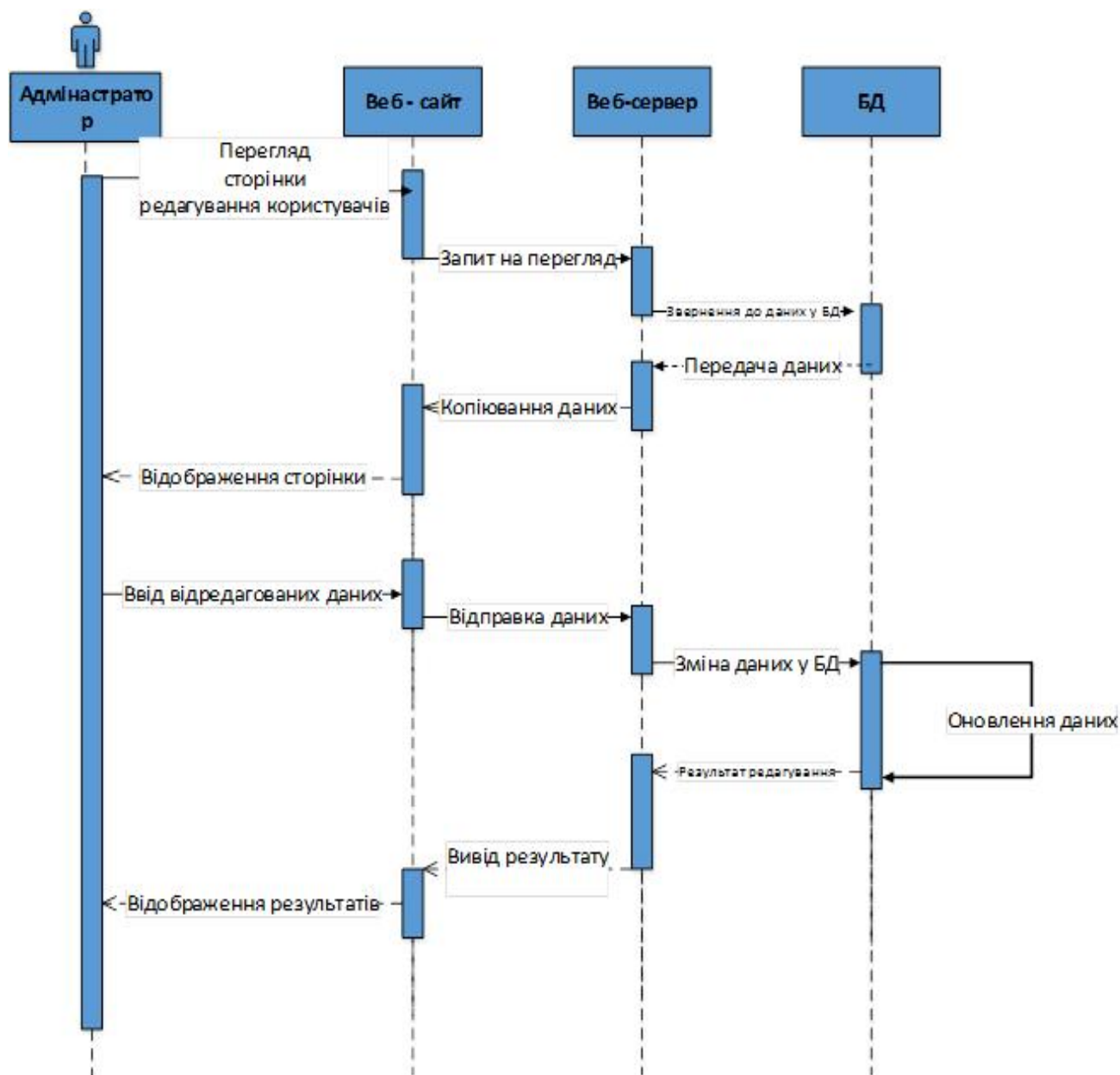


Рисунок 2.3 - Діаграма послідовності (процесу редагування користувачів)

На рисунку 2.4 зображена діаграма послідовності для варіанту використання «Додавання студента», яка відображає взаємодію об'єктів впорядкованих за часом.

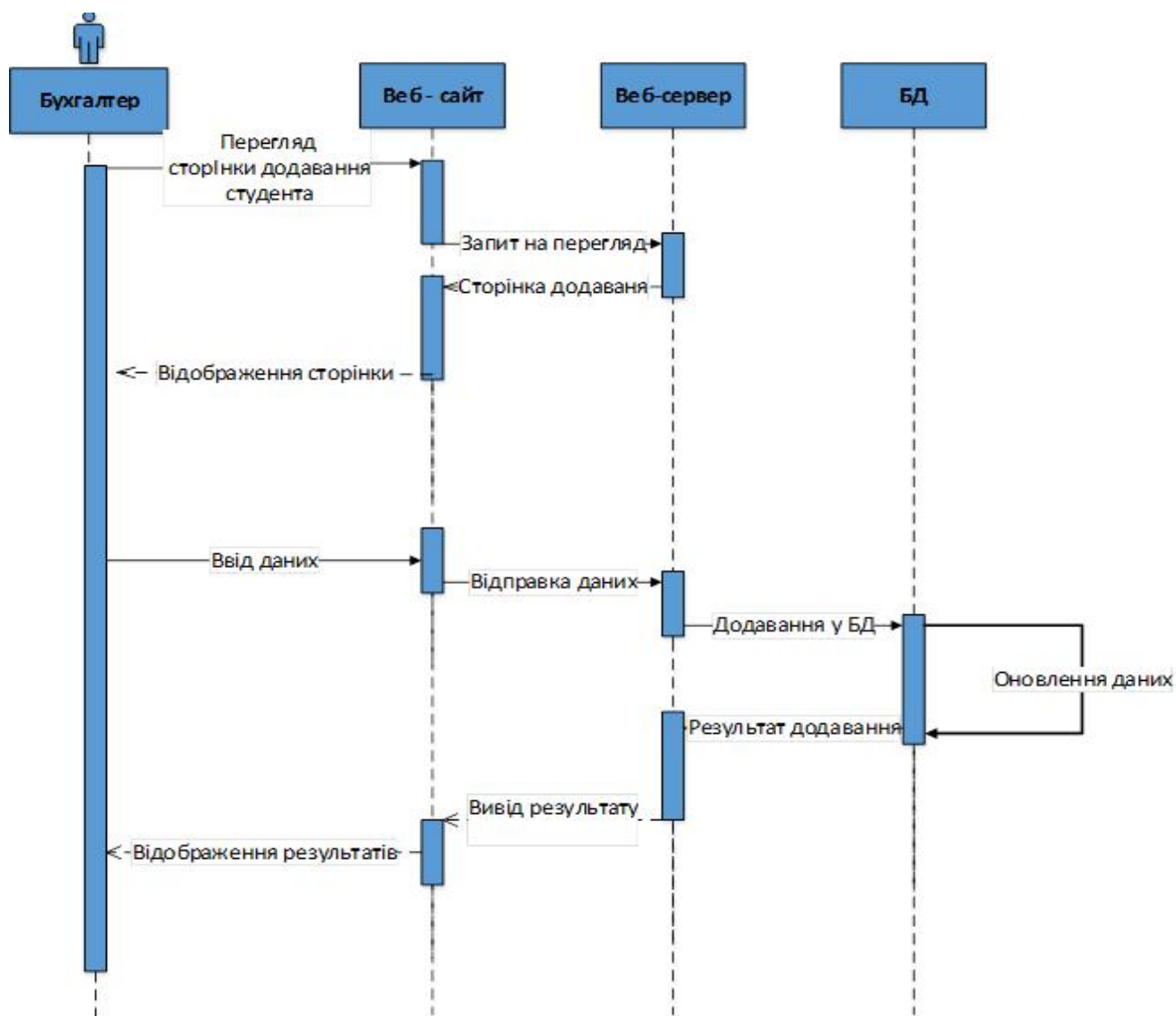


Рисунок 2.4 - Діаграма послідовності процесу додавання студента

Як видно з рисунку, найдовшим процесом буде відображення результатів після внесення змін, оскільки він містить найбільше етапів.

Для моделювання поведінки об'єктів системи при переході з одного стану в інший використовують діаграми станів. Діаграма станів для функції «Редагування» зображено на рисунку 2.5

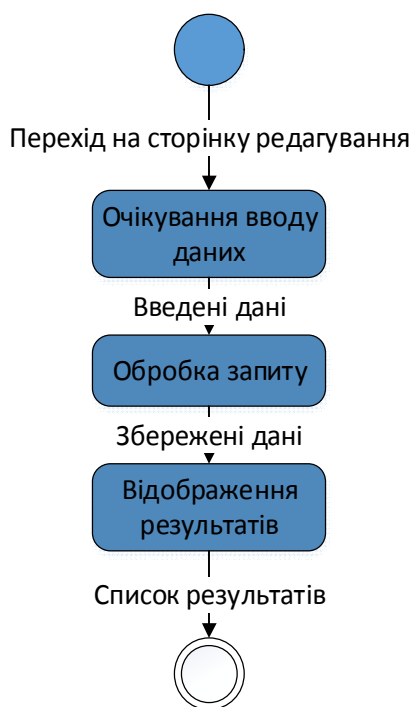


Рисунок 2.5 - Діаграма станів (редагування)

Як видно з рисунку, після переходу на сторінку редагування, система очікує вводу даних. Коли користувач здійснить дану операцію, введені дані обробляються, після чого зберігаються у БД. В подальшому дані оновлюються і відображаються користувачу.

На рисунку 2.6 зображено діаграму станів (видалення)

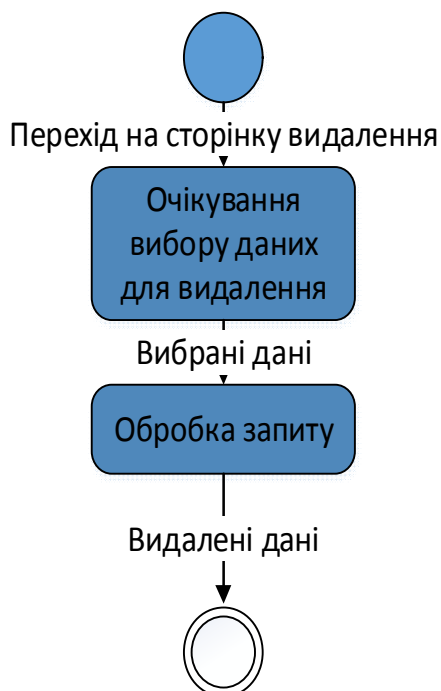


Рисунок 2.6 - Діаграма станів (видалення)

Як видно з рисунку, після переходу на сторінку видалення, обраний кортеж обробляється системою. В ході виконання запиту запис видаляється.

На рисунку 2.7 зображено діаграму станів (додавання студента)

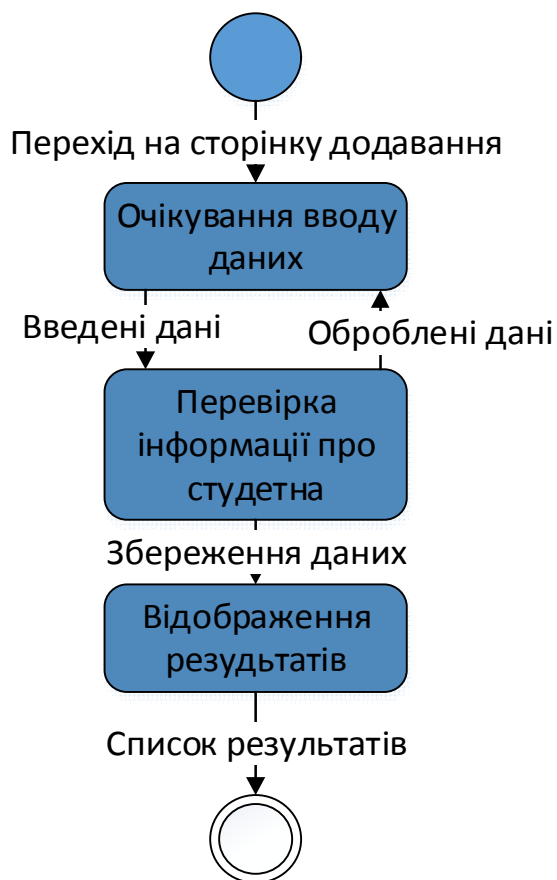


Рисунок 2.7 - Діаграма станів (Додавання студента)

Як видно з рисунку, після переходу на сторінку додавання, система очікує вводу даних. Якщо введено ідентифікаційний код студента, ідентичний тому, який вже є у БД, система повідомить про те, що запис про студента уже присутній і запропонує перевірити правильність вводу. Якщо ж всі поля заповнені коректно дані успішно будуть внесені у БД та відобразиться повідомлення, про успішно доданий запис.

3.2 Проектування структури бази даних

Для нормальної роботи системи, потрібно забезпечити збереження великого обсягу даних, доступ до яких буде можливий лише користувачам

цього програмного продукту. Це стало можливим за допомогою баз даних та засобами їх управління, зокрема системою управління базами даних (СУБД) MySQL, яка використовується в системі.

Для реалізації програмного продукту було обрано клієнт-серверну архітектуру. У цій архітектурі запити розподілені між постачальниками послуг (серверами) та споживачами (користувачами). Фізично клієнт і сервер – це програмне забезпечення. Зазвичай вони між собою взаємодіють, за допомогою мережі, через мережеві протоколи і знаходяться на різних обчислювальних машинах. Програми – сервера, очікують від клієнтських програм запитів і надають їм свої ресурси, у вигляді даних або ж сервісних функцій.

На рисунку 2.8 наведено об'єкти та зв'язки між ними, які будуть використовуватись на етапі експлуатації системи.

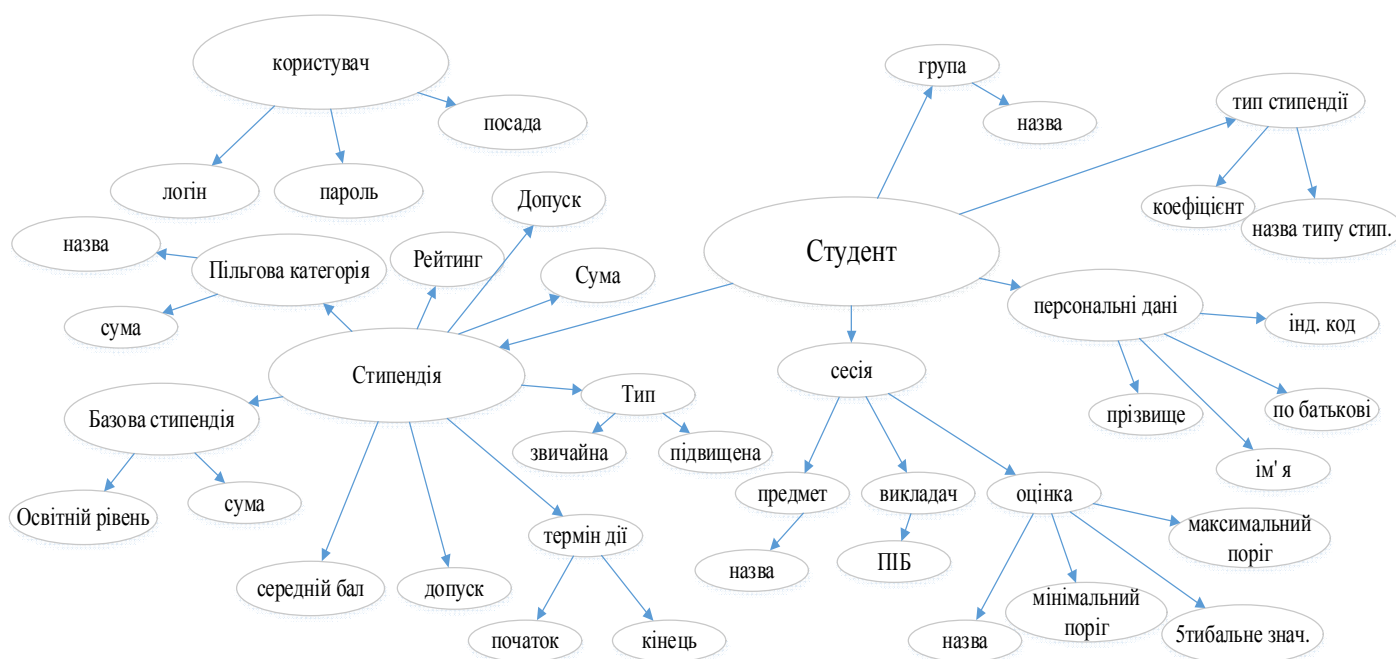


Рисунок 2.8 – Діаграма елементів і зв'язків

При високорівневому (концептуальному) проектуванні баз даних використовують ER-модель[9]. Під час проектування БД відбувається перетворення ER-моделі в конкретну схему бази даних на основі обраної моделі даних. На рисунку 2.9 зображено модель ERD для розроблюваної системи.

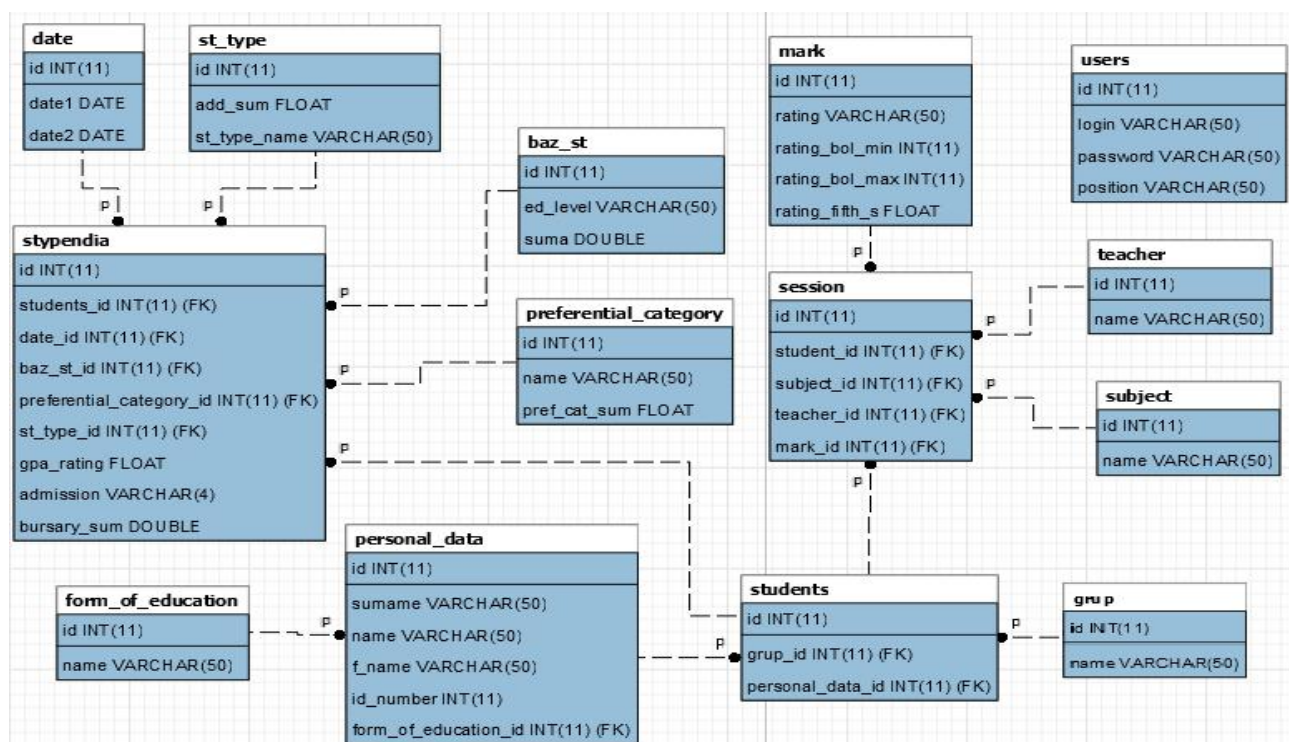


Рисунок 2.9 – Модель сутність-зв'язок (ERD)

Усі відношення автоматично індексуються, по первинному ключі. Завдяки чому значно пришвидшується процес пошуку даних у БД. У розроблюваній системі створено foreign keys для наступних атрибутів:

1. students_id;
2. date_id;
3. baz_st_id;
4. preferential_category_id;
5. st_type_id;
6. form_of_education_id;
7. grup_id;
8. personal_data_id;
9. student_id;
10. subject_id;
11. teacher_id;
12. mark_id.

Результатом фізичного проектування є DDL код, який надано в додатку Б.

Висновки до розділу II

В ході виконання розділу було розглянуто детальну модель веб-орієнтованої системи нарахування стипендії та основні функції, що забезпечуватимуть чітке виконання вимог до системи. Визначено зв'язки між компонентами системи та їх взаємодію. Обрано архітектуру для реалізації веб-орієнтованої системи нарахування стипендії. Розглянуто основні класи, за допомогою яких буде можливим функціонування системи та засоби які будуть реалізовувати бізнес-логіку предметної області. Окрім цього спроектовано схему бази даних та обрано поля для пошуку.

РОЗДІЛ III

ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Програмна реалізація проекту

Обґрунтування вибору мови програмування

Для реалізації веб-орієнтованої програмної системи нарахування стипендії було обрано мову програмування PHP[10].

PHP (Hypertext PreProcessor) – широко-застосовувана мова опису сценаріїв загального призначення із відкритим вихідним кодом. Інакше кажучи, PHP – це мова програмування яка призначена для написання веб-додатків, які будуть виконуватись на веб-сервері. Невід’ємною перевагою PHP є надання можливості веб-розробнику швидкого створення динамічних веб-сторінок. За допомогою цієї мови програмування можна створювати якісні веб-додатки за короткі проміжки часу, які будуть підтримуватись у майбутньому і будуть мати можливість до швидкої модифікації. Окрім цього за допомогою PHP можна досить просто організувати комунікацію із БД. PHP не викликає труднощів у вивченні і водночас здатна задовільнити запити професійних програмістів.

До основних аспектів, які сприяли вибору мови програмування належать:

- легкість у вивченні;
- безкоштовне поширення із відкритим вихідним кодом;
- простота у комунікації із БД;
- добре організована система комунікації із серверами;
- висока швидкість обробки сценаріїв;
- надійно реалізований механізм безпеки;
- гнучкість у використанні;

Організація інтерфейсу з користувачем

При проектуванні інтерфейсу системи основна увага приділялася простоті та легкості сприйняття. Завдяки цьому у користувача який буде вперше працювати із системою не виникне ніяких труднощів, оскільки все буде зрозуміло на інтуїтивному рівні.

В залежності від займаної посади користувачу системи буде відкрито доступ до тих функцій, виконання яких його зобов'язують посадові обов'язки. Інтерфейс головної сторінки системи містить у собі такі блоки:

- Menu:
- Header
- Content
- Footer

Для забезпечення зручності і досягнення максимальної швидкодії в роботі було вирішено використати AJAX[11]. AJAX (Asynchronous JavaScript And XML) - підхід до побудови інтерфейсу, в розгортанні роботи якого веб-сторінка без перезавантаження відсилає певні запити на сервер і повертає потрібні дані. Тобто блоки: menu, header та footer залишаються статичними, а в блок content завантажуються дані, відповідно до обраної користувачем функції. Лістинг коду AJAX наведено нижче.

```
<script>
function showContent(link) {
var cont = document.getElementById('contentBody');
var loading = document.getElementById('loading');
cont.innerHTML = loading.innerHTML;
var http = createRequestObject();
if( http ) {
http.open('get', link);
http.onreadystatechange = function () {
if(http.readyState == 4) {
cont.innerHTML = http.responseText;
}
}
}
```



```

http.send(null);
} else {
document.location = link;
}
}

function createRequestObject() {
try { return new XMLHttpRequest() }
catch(e) {
try { return new ActiveXObject('Msxml2.XMLHTTP') }
catch(e) {
try { return new ActiveXObject('Microsoft.XMLHTTP') }
catch(e) { return null; }
}
}
}
}
</script>

```

Коли в блок content не завантажено великої кількості даних, основне меню залишається без змін. В іншій ситуації, коли в цьому блоці знаходиться великий об'єм даних і є потреба в прогортанні сторінки вниз – блок menu залишається у верхній частині сайту. Це забезпечує додаткову користувацьку свободу, оскільки з любого місця можна повернутися на головну сторінку чи викликати будь-яку іншу функцію. Лістинг коду цієї функції наведено нижче.

```

$(document).ready(function(){
$("#header").removeClass("default");
$(window).scroll(function(){
if ($(this).scrollTop() > 20) {
$("#header").addClass("default").fadeIn('fast');
} else {
$("#header").removeClass("default").fadeIn('fast');
};
});
$('a[href^="#"]').click(function () {
elementClick = $(this).attr("href");

```

```
destination = $(elementClick).offset().top;
if($.browser.safari){
$('body').animate( { scrollTop: destination }, 1000 );
} else {
$('html').animate( { scrollTop: destination }, 1000 );
}
return false;
});
});
```

Вимоги до апаратного забезпечення

- Персональний комп'ютер;
- Встановлений інтернет-браузер;
- Доступ до мережі інтернет;

Опис класів

При проектуванні веб-орієнтованої системи нарахування стипендії використовувався об'єктно орієнтований підхід. Суть цього підходу полягає у представленні програми у вигляді об'єктів, які певним чином взаємодіють між собою. Об'єктом являється екземпляр класу, а клас – певний спосіб опису сутностей, який безпосередньо визначає правила для взаємодії, поведінку та стан.

В процесі розробки програмної системи було реалізовано такі класи:

- Site
- Loader
- Students
- Stypendia
- Users
- Report

За допомогою класу `Site`, методу `go()` відбувається процес ідентифікації користувача та надання йому відповідного рівня доступу до системи. Якщо користувача із введеними даними не зареєстровано у системі – в доступі до основного функціоналу буде відмовлено.

За допомогою класу `Loader` відбувається побудова блоків сайту та механізм їх завантаження. Цей клас містить наступні методи:

- `loadScript();`
- `showBlocks();`
- `authBuild();`
- `menuBuild();`
- `contentBuild();`
- `footerBuild();`

За допомогою класу `Students` можна виконувати перегляд списку студентів, додавання нового студента, редагування даних студента, та видалення запису із БД. Окрім цього є можливість формування списків студентів за певними критеріями, а саме: група та форма навчання. Крім цього реалізований також пошук по прізвищу. Цей клас містить наступні методи:

- `openStudents();`
- `searchMenuBuild();`
- `searchResult();`
- `addStudent();`
- `updateStudents();`
- `deleteStudent();`

За допомогою класу `Stypendia` можна обрахувати середній бал студентів та призначити стипендію відповідно до певних критеріїв, а саме: можливості отримувати пільгову та підвищену стипендію, редагувати базові відомості про стипендію (дату дії, розмір підвищеної стипендії, розмір звичайної стипендії, відповідно до ОКР). Цей клас містить наступні методи:

- `CalculateGPA_and_bursarySum();`
- `updateDate();`

- update_st_type();
- update_baz_st();

Вихідний код, який реалізовується в методі CalculateGPAandBursarySum зображено нижче.

```
function
CalculateGPA_and_bursarySum($load_data_query,$max_students_id_query,$update_GPA_query)
{
$max_students_id_query =(" SELECT MAX(students_id) as m_s_id FROM stypendia ");
$res = mysql_query($load_data_query) or die(mysql_error());
$max_student_id = mysql_query($max_students_id_query) or die(mysql_error());
while ($m_s_id_arr = mysql_fetch_array($max_student_id)) {
echo "<tr>\n";
$m_s_id = $m_s_id_arr['m_s_id'];
for ($i=1; $i<=$m_s_id;$i++){
$update_GPA_query =("UPDATE stypendia
SET gpa_rating = (
Select avg(mark.rating_fifth_s)
FROM mark
INNER JOIN `session` ON `session`.mark_id = mark.id
WHERE `session`.student_id = '$i'
)
WHERE stypendia.students_id ='$i'
");
$res3 = mysql_query($update_GPA_query) or die(mysql_error());
}
}
echo (
<table class="simple-little-table" border="1" cellspacing="0">
<tr style="border: solid 1px #000">
<td align="center"><b>Група</b></td>
<td align="center"><b>Прізвище</b></td>
<td align="center"><b>Ім*я</b></td>
<td align="center"><b>По батькові</b></td>
<td align="center"><b>Пільгова категорія</b></td>
<td align="center"><b>Середній бал</b></td>
```

```

<td align="center"><b>Допуск</b></td>
<td align="center"><b>Сума, грн</b></td>
<a id ="1">
</tr>
');
while ($row = mysql_fetch_array($res)) {
echo "<tr>\n";
echo "<td>".$row['grup'].</td>\n";
echo "<td>".$row['surname'].</td>\n";
echo "<td>".$row['student_name'].</td>\n";
echo "<td>".$row['f_name'].</td>\n";
echo "<td>".$row['preferential_category'].</td>\n";
echo "<td>".$row['gpa_rating'].</td>\n";
echo "<td>".$row['admission'].</td>\n";
echo "<td>".$row['bursary_sum'].</td>\n";
}
echo ("</table>\n </a>");
$res = mysql_query('CALL `calculate_stypendia`()') or die(mysql_error());
$max_student_id2 = mysql_query($max_students_id_query) or die(mysql_error());
while ($m_s_id_arr2 = mysql_fetch_array($max_student_id2)) {
$m_s_id2 = $m_s_id_arr2['m_s_id'];
for ($j=1; $j<=$m_s_id2;$j++){
$update_bursary_sum_query =("UPDATE stypendia
SET bursary_sum = (
SELECT `sum` FROM (
SELECT suma + pref_cat_sum + add_sum as `sum`
FROM stypendia
INNER JOIN baz_st on baz_st.id = stypendia.baz_st_id
INNER JOIN preferential_category on preferential_category.id =
stypendia.preferential_category_id
INNER JOIN st_type on st_type.id = stypendia.st_type_id
WHERE stypendia.students_id = '$j'
)
AS ResultSet
)

```

```

WHERE stypendia.students_id ='$j' AND stypendia.admission = 'yes'
");
$resultat = mysql_query($update_bursary_sum_query) or die(mysql_error());
}
}
mysql_close();
}

```

За допомогою класу `Users` можна переглянути список зареєстрованих користувачів системи, редагувати дані користувачів, додати нового та видалити уже існуючого користувача системи. Цей клас містить наступні методи:

- `openUsers();`
- `addUser();`
- `updateUsers();`
- `deleteUser();`

Клас `Report` дозволяє сформувати наказ на стипендію та переглянути звітність по стипендії. Є можливість формування списку по певним критеріям (група, наявність пільгової категорії, рейтинг). Окрім цього вкінці списку буде вказано кількість результатів, які задовольняють вище-згадані критерії, загальний середній рейтинг по цих результатах та загальна потрачена сума. Цей клас реалізує метод `ReportSearchResult()`.

3.2 Програмна реалізація бази даних

Для комунікації із системою було обрано СУБД MySQL[12]. На сьогоднішній день MySQL набула широкого застосування і підтримується великою кількістю мов програмування. Основним застосунком якої є керування реляційними БД (тих баз даних, у яких дані зберігаються в певних таблицях завдяки чому забезпечується максимальна швидкість та гнучкість). Середовищем розробки і адміністрування БД MySQL було обрано MySQL Workbench, оскільки він є безкоштовним та зручним у використанні.

До основних переваг MySQL Workbench можна віднести:

- можливість редагування даних в таблиці, а також можливість легко представити графічну модель БД;
- наявність налагодженого і простого в застосуванні механізму створення зв'язків між таблицями;
- наявність простого редактора SQL-запитів, в результаті виконання яких отримаємо таблицю із відповідними даними;

Основним інструментом для операцій з таблицями використано SQL-запити[13]. Існує 4 типи SQL-запитів, а саме:

- вибірка (SELECT *назва_колонки_1, назва_колонки_2* FROM *назва_таблиці*);
- вставка (INSERT INTO *назва_таблиці* (*назва_колонки_1, назва_колонки_2, назва_колонки_3, ...*) VALUES (*значення_1, значення_2, значення_3, ...*));
- редагування (UPDATE *назва_таблиці* SET *назва_колонки_1 = значення_1, назва_колонки_2 = значення_2, ...* WHERE *деяка_колонка = деяке_значення*);
- видалення (DELETE FROM *назва_таблиці* WHERE *деяка_колонка = деяке_значення*);

Для реалізації бізнес-логіки процесу нарахування стипендії в методі CalculateGPAandBursarySum() класу Stypendia відбувається виклик збереженої процедури під назвою calculate_stypendia. Вихідний код stored-процедури[14] наведено нижче.

```
CREATE DEFINER=`root`@`%` PROCEDURE `calculate_stypendia`()
LANGUAGE SQL
NOT DETERMINISTIC
MODIFIES SQL DATA
SQL SECURITY DEFINER
COMMENT ''
BEGIN
UPDATE stypendia
SET stypendia.st_type_id = (SELECT id from st_type where
st_type.st_type_name = 'Звичайна')
WHERE stypendia.gpa_rating >=4 and stypendia.gpa_rating<5;
UPDATE stypendia
SET stypendia.st_type_id = (SELECT id from st_type where
st_type.st_type_name = 'Підвищена')
WHERE stypendia.gpa_rating =5;
UPDATE stypendia
SET stypendia.st_type_id = (SELECT id from st_type where
st_type.st_type_name = 'Звичайна')
WHERE stypendia.gpa_rating >=3 and stypendia.gpa_rating<4 and
stypendia.preferential_category_id != (SELECT id from
preferential_category WHERE preferential_category.name='Немає');
UPDATE stypendia
```

```
SET stypendia.admission = 'no'
WHERE stypendia.gpa_rating <4;
UPDATE stypendia
SET stypendia.admission = 'yes'
WHERE stypendia.gpa_rating >=4;
UPDATE stypendia
SET stypendia.admission = 'yes'
WHERE stypendia.preferential_category_id != (SELECT id from
preferential_category WHERE preferential_category.name='Немає');
UPDATE stypendia
SET stypendia.admission = 'no'
WHERE stypendia.gpa_rating <3;
UPDATE stypendia
SET stypendia.bursary_sum = 0
WHERE stypendia.admission = 'no';
END
```

Висновки до розділу III

В процесі виконання розділу було обґрунтовано вибір мови програмування яка використовується, розглянуто основні аспекти проектування інтерфейсу з користувачем. Обрано підхід, за допомогою якого користувацький інтерфейс досягатиме максимальної швидкодії в роботі. Визначено вимоги до апаратного забезпечення. Проаналізовано основні класи, які використані при проектуванні веб-орієнтованої системи нарахування стипендії та наведено приклад їх програмної реалізації. Також обрано основні інструменти, необхідні для комунікації із системою та наведено їхні основні переваги.

РОЗДІЛ IV

ТЕСТУВАННЯ ТА ДОСЛІДНА ЕКСПЛУАТАЦІЯ

4.1 Тестування

Тестування – завершальний етап у розробці будь-якого програмного продукту, який відіграє дуже важливу роль в створенні якісного ПЗ. З рівнем складності програми зростає і рівень складності її тестування. Основною метою тестування є класифікація рівня якості програмного продукту. Ця задача реалізовується за допомогою двох основних чинників, а саме:

- демонстрація учасникам проекту відповідності програми до поставлених вимог;
- визначення факторів, в ході яких певні функції системи різняться від тих, що прописані в специфікації.

Практично неможливо зразу ж написати ідеальний програмний продукт який б не містив у собі ніяких помилок. І як буває у більшості випадків, винуватцями появи помилок у розроблюваному програмному продукті є самі програмісти. Отож задачею кваліфікованого програміста є не лише написання програми, а й відстеження в ній помилок. Виходячи з цього тестування веб-орієнтованої системи нарахування стипендії є не лише доцільним, а й необхідним етапом розробки.

Під час проведення тестування програмного продукту були використані такі види тестування[15]:

- функціональне тестування;
- тестування сумісності;
- тестування продуктивності;
- тестування GUI;
- тестування безпеки.

Функціональне тестування

В розгортанні функціонального тестування перевірятимуться усі функції системи на предмет відмінності між реальною поведінкою і очікуваною, відповідно до специфікації.

Використано як ручне тестування так і автоматизоване. Інструментом для реалізації автоматизованого тестування слугував плагін для веб-браузера Mozilla Firefox під назвою Selenium[16].

На рисунку 4.1 зображено результат тестування у Selenium функції додавання студента.

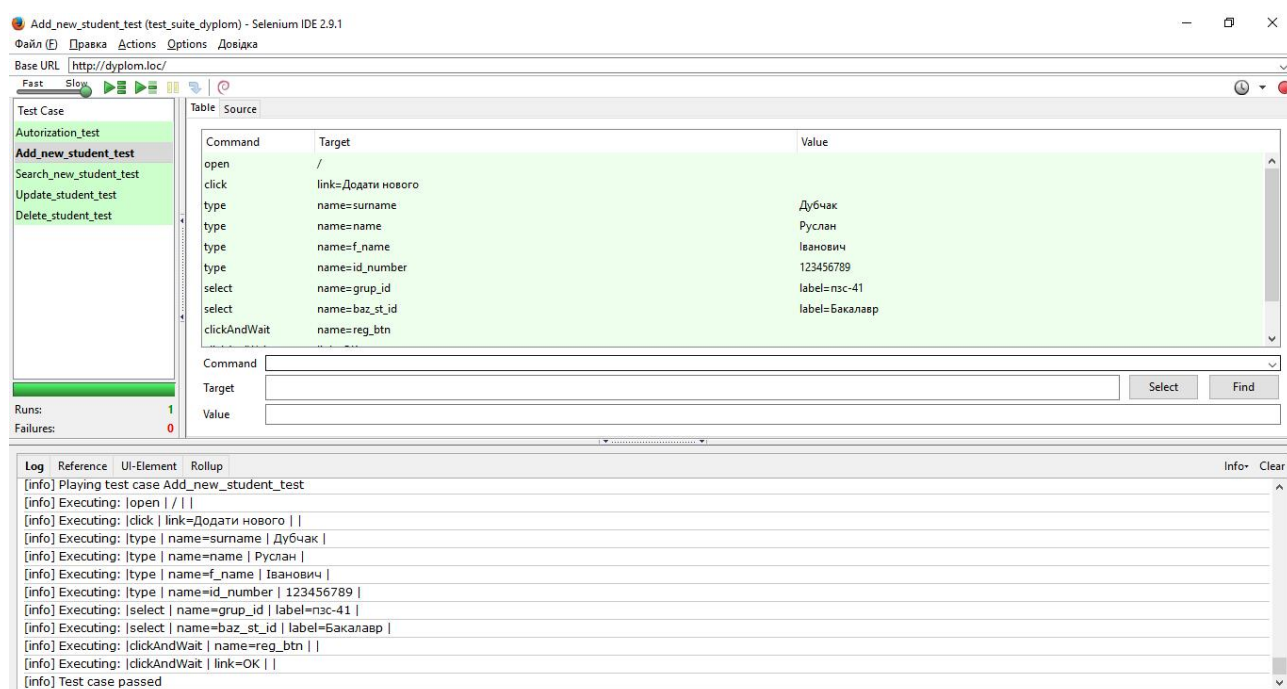


Рисунок 4.1 – Тестування функції додавання студента у Selenium

Перелік, вихідний код та логи тест-кейсів[17] наведено у додатку В.

В ході розробки системи та її тестування було налаштовано та наповнено тестову базу даних.

Під час розробки тестових випадків було спроектовано тести для функціонального тестування, тестування безпеки. Кожен тестовий випадок містить детальні кроки, тестові дані і очікувані результати. Спроектовано 10 функціональних тестових випадків. Результат наведено у додатку Г.

В таблиці 4.1 наведено результати розподілу функціональних тестових випадків та наборів текстових даних для них за варіантами використання, на прикладі:

- варіанту використання «авторизація»;
- варіанту використання «додавання студента»;
- варіанту використання «перегляд списку студентів»;
- варіанту використання «пошук студента»;
- варіанту використання «редагування даних студента»;
- варіанту використання «видалення даних студента»

Таблиця 4.1

Розподіл функціональних тестових випадків та наборів текстових даних для цих випадків за варіантами використання

Варіанти використання	Тестові випадки	Тестові дані
Check Authorization	3	3
Check option "Add_student"	5	5
Check option "view_students"	5	5
Check option "view_students_search_result"	3	3
Check option "update_students"	5	5
Check option "delete_student"	5	5
Загалом	26	26

Тестування сумісності

Тестування сумісності – вид тестування метою якого являється перевірка коректності роботи програмного продукту в різних середовищах, в нашому випадку середовищем виступає веб-браузер. Завдяки цьому тесту отримаємо змогу відслідкувати і виправити недоліки системи задля правильного функціонування у всіх браузерах.

Перевірка сумісності буде проводитись у таких браузерах:

- Google Chrome;

- Opera;
- Mozilla Firefox;
- Internet Explorer;
- Microsoft Edge;

Підсумок тестування: В результаті тестування пройшли успішно 99 зі 100 множин тестових даних, отже тестування на сумісність розглядається як частково успішне адже 99% тестових випадків пройшли.

Відомі дефекти:

Дефект: В браузері Mozilla Firefox при перегляді сторінки зі звітністю, блок вибору критеріїв пошуку змістився вліво.

Опис: при перегляді сторінок в різних браузерах інтерфейс системи повинен залишатись однаковим, допускається лише мінімальне відхилення.

Подолання дефекту: Такого роду дефект можна виправити, змінивши тип позиціонування блоку з “align = center” на певне значення rx відступу зліва.

Тестування продуктивності

Основними показниками продуктивності програмного продукту є час з'єднання з сервером та час відповіді сервера. Чим менше значення цього показника тим краще, адже запити будуть швидко обробляться і виконуватись.

Ці показники можна дізнатись за допомогою тестування продуктивності, в процесі виконання якого, шляхом навантаження буде визначено їх швидкодію.

Для перевірки продуктивності системи використано такі види тестування продуктивності:

- тестування стабільності;
- конфігураційне тестування;
- навантажувальне тестування;
- стрес-тестування.

Підсумок тестування: В результаті тестування 10 із 10 тестів виявились успішними. Час з'єднання з сервером – 0,12 с., час відповіді сервера – 0,23 с. Відповідно, виходячи із отриманих результатів, тестування можна вважати успішним.

Тестування GUI

Тестування GUI (graphical user interface) [18] – вид тестування, метою якого являється визначення степені зручності певного об'єкта (користувацького інтерфейсу, веб-сторінки) для його безпосереднього застосування. Інакше кажучи - це перевірка на ергономічність. Тобто свого роду оцінка зручності використання програми шляхом залучення звичайних користувачів в якості тестувальників і аналіз їх висновків.

В ході тестування GUI було залучено 5 користувачів, яким запропоновано протестувати інтерфейс системи.

Підсумок тестування: 5 з 5 користувачів, які займались тестуванням GUI одногolosно стверджували про те, що інтерфейс системи є легким для сприйняття та зрозумілим на інтуїтивному рівні. Виходячи з цього можна зробити висновок, що тестування GUI веб-орієнтованої системи нарахування стипендії виявилось вдалим.

Тестування безпеки

Тестування безпеки – вид тестування метою якого являється визначення ступеня захисту даних програмного продукту при збереженні функціональності.

Основні аспекти безпеки, з яких складається тестування:

- конфіденційність;
- цілісність;
- аутентифікація;
- доступність даних;

- безвідмовність.

Підсумок тестування:

В результаті тестування 10 із 10 тестів виявились успішними. Доступ до функціоналу системи сайту можливий після проходження процесу авторизації в результаті проходження якого відкриється функціонал, відповідно до типу облікового запису. Паролі для доступу в систему зберігаються під MD5-кодуванням. Завдяки цьому забезпечується додатковий рівень безпеки, оскільки вгадати чи декодувати пароль під цим типом шифрування практично неможливо. Виходячи з отриманих даних можна зробити висновок, що тестування безпеки пройшло успішно.

4.2 Розгортання програмного продукту

Щоб мати змогу користуватися веб-орієнтованою системою нарахування стипендії її потрібно попередньо розмістити на сервері. Список вимог, дотримання яких необхідно для перетворення звичайного ПК в сервер:

- Статичний IP[19];
- Мінімальна швидкість доступу до мережі інтернет 10Mbps;
- Рекомендовано встановлення Unix-системи;
- Встановлення і конфігурація Apache-сервера, PHP, MySQL, PHPMyAdmin та FTP-сервера ;
- Відкриття 80-го порта на вхід в Firewall[20];
- Постійно включений комп'ютер

4.3 Інструкція користувача

Для відкриття основного функціоналу системи потрібно авторизуватися. Цей процес передбачає введення логіна та пароля і натискання на кнопку «Вхід».

На рисунку 4.2 зображено сторінку авторизації.

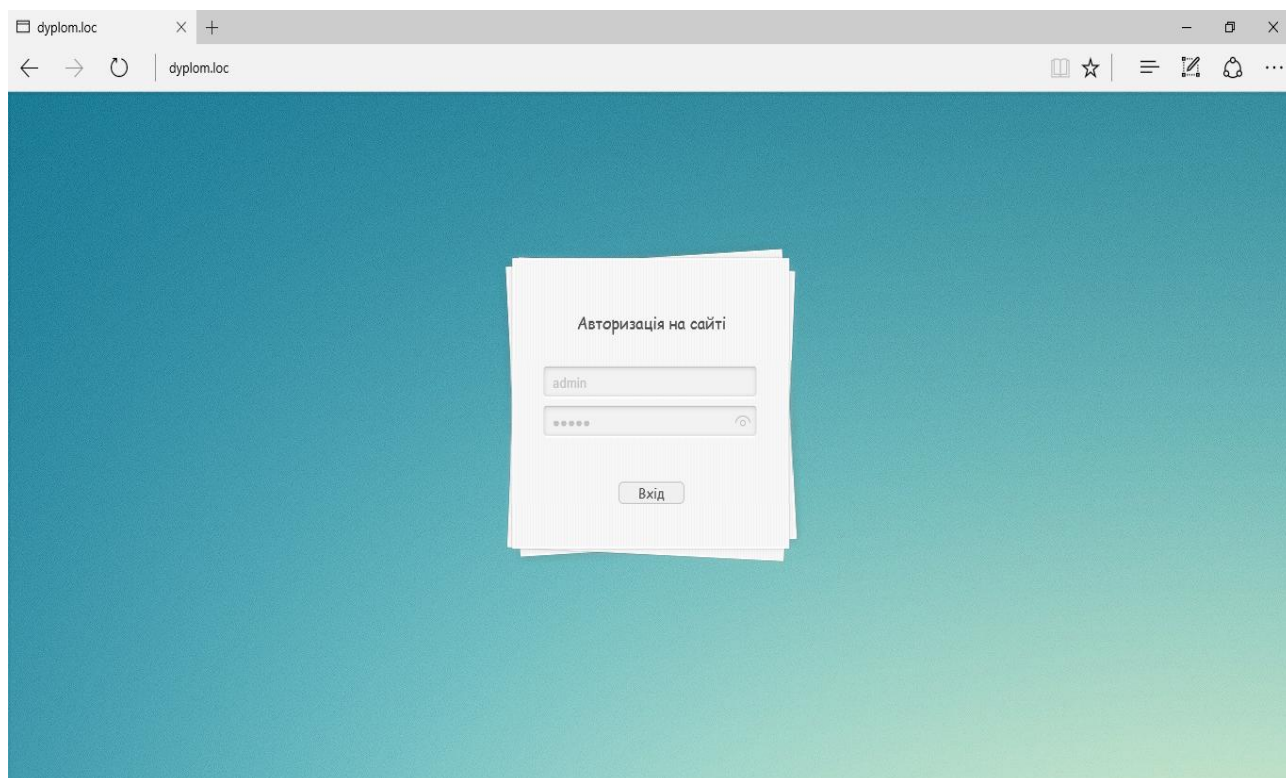


Рисунок 4.2 – Авторизація в системі

Після введення коректних даних, система відповідно до рівня доступу користувача відкриє основний функціонал. Також відповідні дані запишуться у змінні сесії, за допомогою яких при відкритті сторінок системи в інших вікнах браузера користувач буде залишатися залогованим у системі. На рисунку 4.3 зображено головну сторінку сайту.

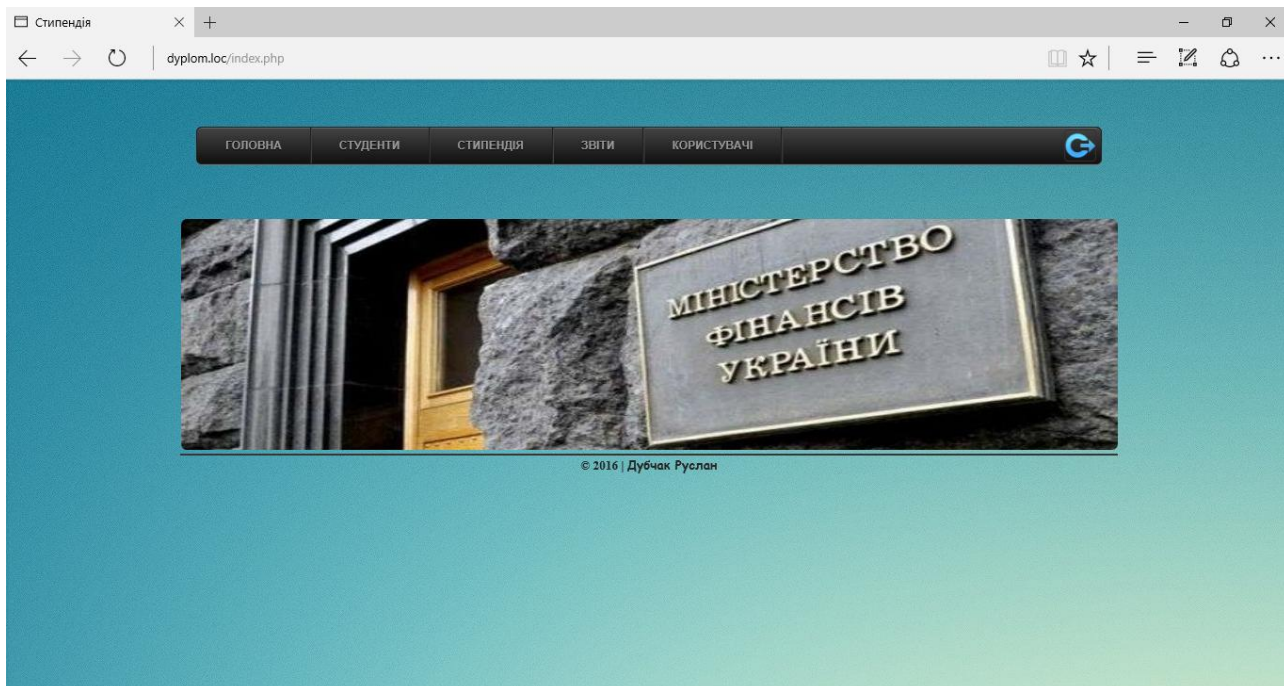


Рисунок 4.3 – Головна сторінка сайту

Для того, щоб обрати певну функцію системи використовується навігаційне меню. Вибір функції на прикладі вкладки «Стипендія -> Редагувати» зображено на рисунку 4.4.

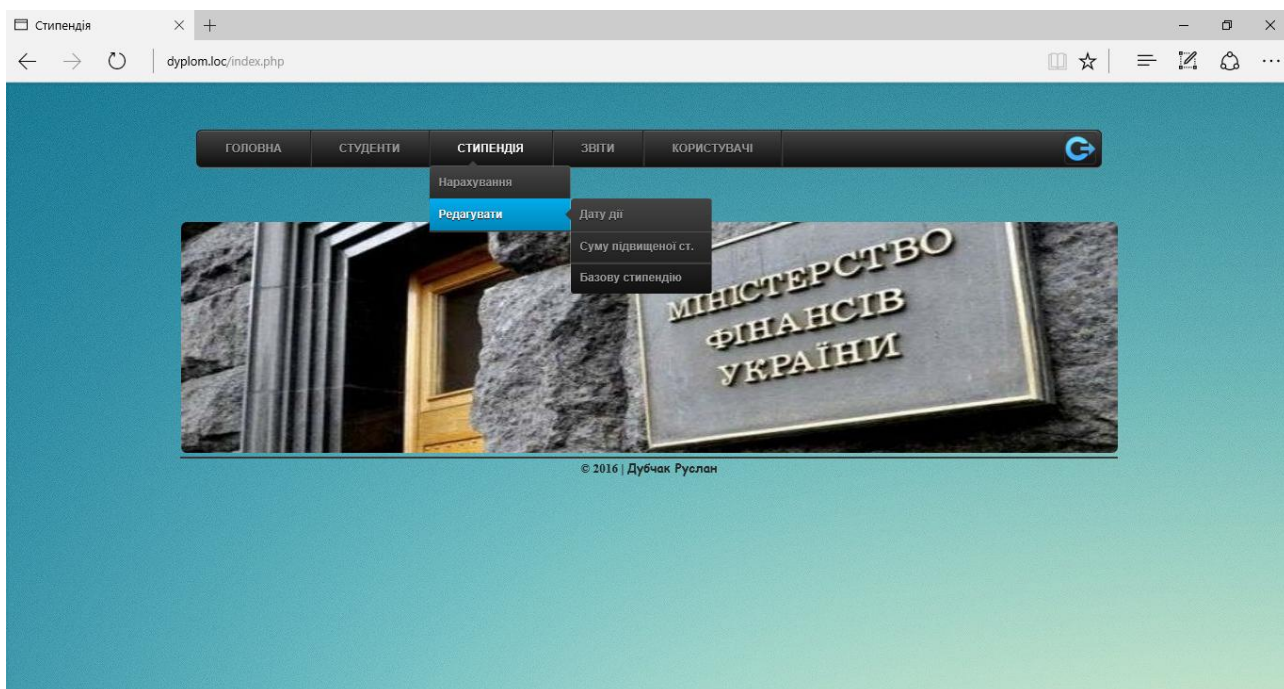
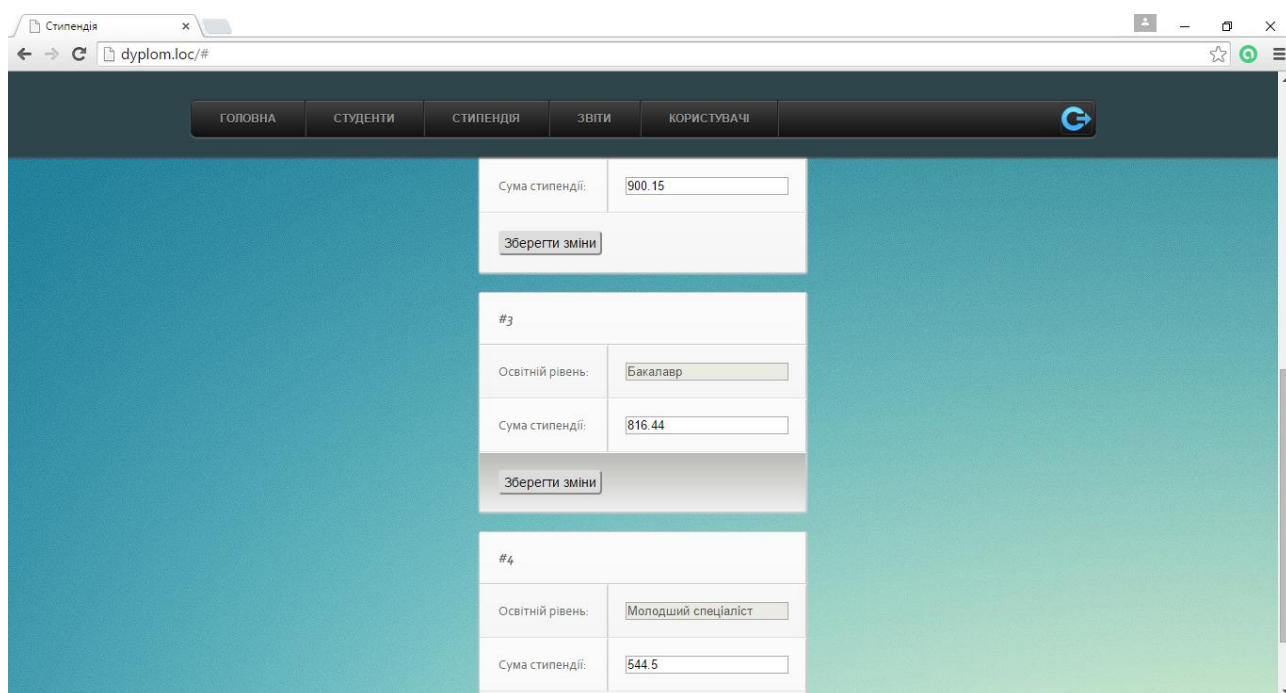


Рисунок 4.4 - Вибір функції «Стипендія -> Редагувати»

Як видно з рисунку 4.4 - наведений вказівник викликає третій рівень меню, за допомогою якого можна редагувати базові стипендійні відомості. Це досить зручно, оскільки редагування базових стипендійних відомостей можна реалізувати працівнику засобами системи, без потреби втручання фахівця і змін вихідного коду. На екранну форму завантажуються відповідні дані із БД, надаючи можливість користувачу системи переглянути теперішні дані та внести зміни (див. рис. 4.5). Після натискання на кнопку «Зберегти зміни» дані успішно перезапишуться у БД. Після цього користувачеві буде повідомлено про успішно здійснену операцію (див.рис. 4.6).



The screenshot shows a web browser window with the URL `diplom.loc/#`. The page has a dark navigation bar with buttons for 'ГОЛОВНА', 'СТУДЕНТИ', 'СТИПЕНДІЯ', 'ЗВІТИ', and 'КОРИСТУВАЧІ'. The main content area is teal and contains three forms for editing scholarship data. Each form has a 'Зберегти зміни' button.

Form ID	Освітній рівень	Сума стипендії
#1	Бакалавр	900.15
#2	Бакалавр	816.44
#3	Молодший спеціаліст	544.5

Рисунок 4.5 – Редагування базової стипендії

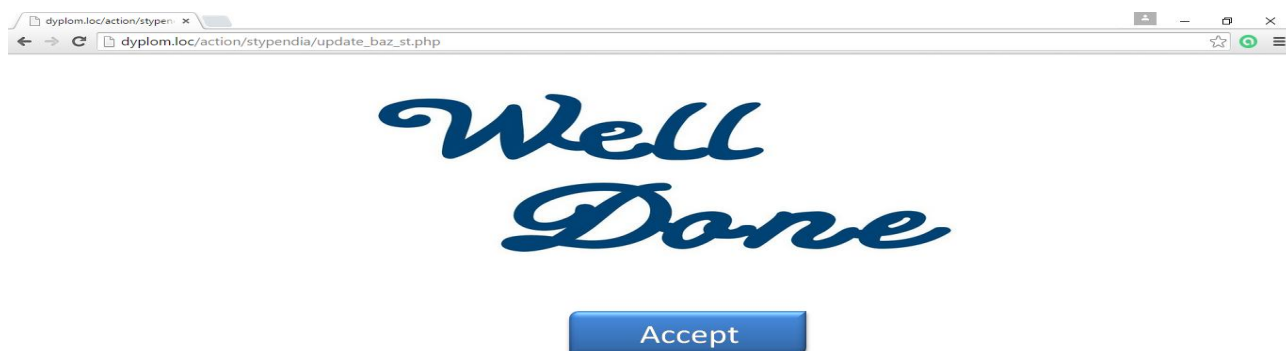


Рисунок 4.6 – Успішно виконана операція

Для обрахунку стипендійних даних необхідно обрати пункт «Стипендія -> Нарахування». Відповідно до результатів заліково-екзаменаційної сесії формується рейтинг та надається допуск для отримання стипендії, відповідно до цього рейтингу та наявності пільгової категорії. Результат цієї дії зображено на рисунку 4.7.

Група	Прізвище	Ім'я	По батькові	Пільгова категорія	Середній бал	Допуск	Сума, грн
pzs-11	Мартиненко	Григорій	Йосипович	Немає	3.5	no	0
pzs-11	Андрусак	Іван	Степанович	Сироти	5	yes	3516.44
pzs-22	Ігоренко	Петро	Андрійович	Жителі гірських місцевостей	4	yes	1700.15
pzs-41	Ковальчук	Іван	Іванович	Немає	4.5	yes	816.44
pzs-41	Дубчак	Руслан	Іванович	Немає	4.5	yes	816.44
pzs-41	Бойко	Микола	Ігорович	Немає	5	yes	916.44

© 2016 | Дубчак Руслан

Рисунок 4.7 – Результат виконання функції «Стипендія -> Нарахування»

Для того, щоб сформувати наказ на стипендію потрібно обрати пункт «Звіти -> Сформувати наказ». На основі раніше обрахованих даних формується список стипендіатів та відомостей про них. Результатом дії цієї функції буде сформований для друку наказ, який зображено на рисунку 4.8.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

НАКАЗ

«Про призначення стипендій»

на __ семестр 20__-20__ н.р.»

Згідно протоколу засідання стипендіальної комісії № __ від __ 20__ року Положення «Про порядок використання коштів, передбачених на виплату стипендій, надання матеріальної допомоги та преміювання студентів Тернопільського національного економічного університету»

НАКАЗУЮ:

Призначити стипендію студентам денної форми, що навчаються за держзамовленням, на __ семестр 20__-20__ н.р., з __ р. по __ р., які зазначені у реєстрі № 1.

РЕЄСТР № 1

Група	Прізвище	Ім'я	По батькові	Пільгова категорія	Середній бал	Сума, грн	Тип стипендії	Дата початку дії наказу	Дата кінця дії наказу
pzs-11	Андрусак	Іван	Степанович	Сироти	5	1516.44	Південна	2016-05-08	2016-12-08
pzs-22	Ігоренко	Петро	Андрійович	Жителі гірських місцевостей	4	1700.15	Звичайна	2016-05-08	2016-12-08
pzs-41	Бойко	Микола	Ігорович	Немає	5	916.44	Південна	2016-05-08	2016-12-08
pzs-41	Ковальчук	Іван	Іванович	Немає	4.5	816.44	Звичайна	2016-05-08	2016-12-08

Ректор, професор А. І. Кривосавитський _____
Головний бухгалтер М.І. Пономаренко _____

Ректор, професор А. І. Кривосавитський

Рисунок 4.8 - Результат дії функції «Звіти -> Сформувати наказ»

Перегляд звітності по призначеній стипендії можна здійснити за допомогою функції «Звіти -> Перегляд», результатом дії якої буде сторінка на якій потрібно обрати критерії формування звіту. Цю сторінку зображено на рисунку 4.9. Після вибору критеріїв формування списків і натискання на кнопку пошуку сформується таблиця із відповідним звітом (див. рис. 4.10).

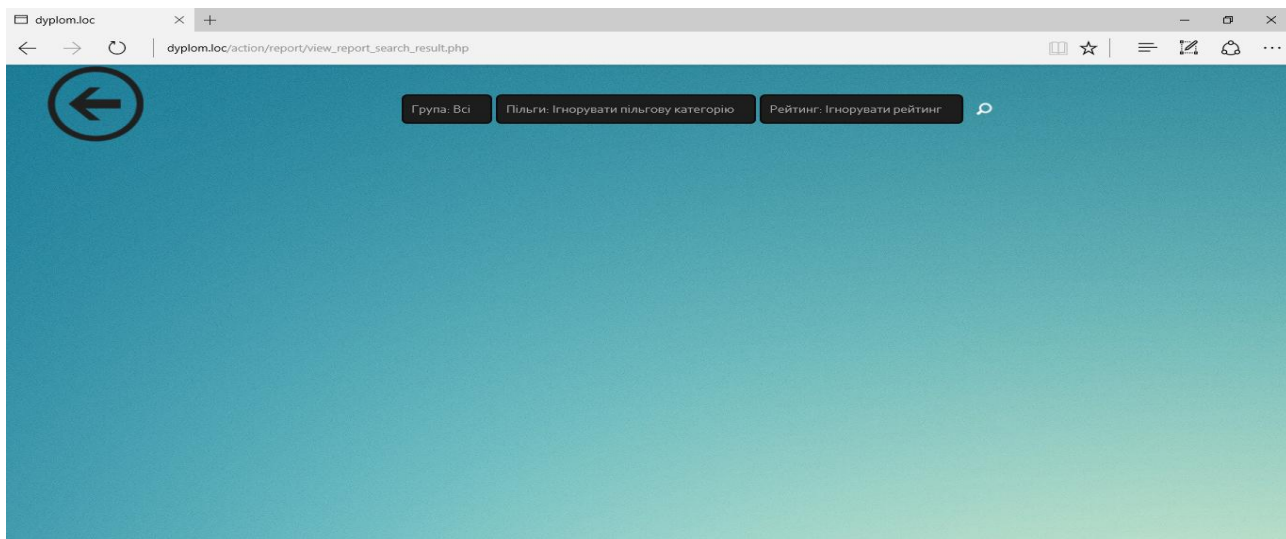


Рисунок 4.9 - Результат дії функції «Звіти -> Перегляд»

Група	Прізвище	Ім'я	по Батькові	Пільгова категорія	Рейтинг	Сума стипендії	Тип стипендії
p25-41	Ігоренко	Петро	Андрійович	Жителі гірських місцевостей	4	1700.15	Звичайна
p25-41	Андрусак	Іван	Степанович	Сироти	5	1516.44	Підвищена
p25-41	Мартиненко	Григорій	Йосипович	Немає	3.5	0	Звичайна
p25-41	Бойко	Микола	Ігорович	Немає	5	916.44	Підвищена
p25-41	Ковальчук	Іван	Іванович	Немає	4.5	816.44	Звичайна

Кількість результатів: 5; Середній рейтинг: 4.40; Загальна сума: 4949.47;

Рисунок 4.10 – Сторінка зі сформованим результатом стипендійних даних по обраним критеріям

Для перегляду списку студентів використовується пункт «Студенти -> Відкрити список». Результатом виконання цієї функції буде сторінка із

сформованим списком всіх студентів, меню вибору сортування та пошуку (див. рис. 4.11).

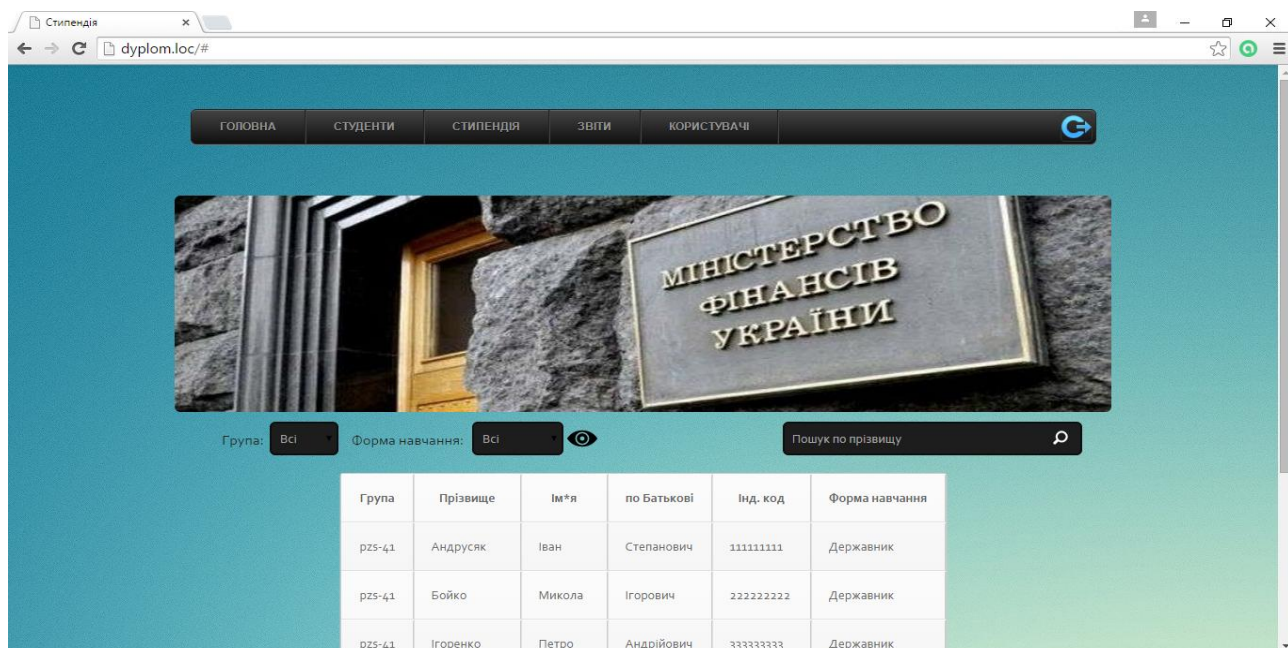


Рисунок 4.11 – Результат виконання функції «Студенти -> Відкрити список»

Для виходу із системи потрібно натиснути на кнопку logout, яка знаходиться у правому куті навігаційного меню (Рисунок 4.12). При наведенні курсора на цю кнопку її піктограма зміниться (Рисунок 4.13) і після натискання здійсниться вихід із системи.

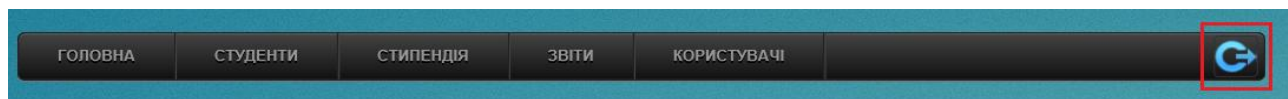


Рисунок 4.12 – Кнопка «logout»

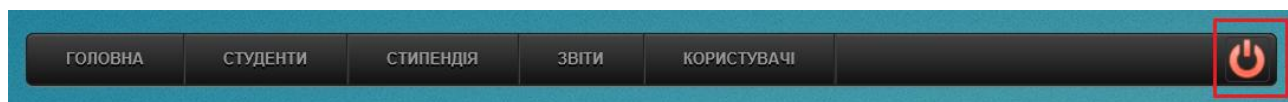


Рисунок 4.13 – Кнопка «logout» при наведенні курсора

Після цієї дії змінні сесії очищуються і сеанс роботи із системою вважається завершеним.

Висновки до розділу IV

В ході виконання розділу було виконано тестування веб-орієнтованої системи нарахування стипендії. Реальна поведінка усіх функцій системи відповідає тій, що прописана у специфікації. Налаштовано та наповнено тестову базу даних. Перевірено роботу програмного продукту у різних браузерах. Визначено час з'єднання та час відповіді сервера. Ідентифіковано ступінь зручності користувацького інтерфейсу та ступень захисту даних програмного продукту при збереженні функціональності.

Також складено інструкцію користувача та список вимог, дотримання яких необхідно для перетворення звичайного ПК в сервер.

Отже, після проведення тестування веб-орієнтованої системи нарахування стипендії можна зробити висновок про те, що загальний результат – позитивний. Оскільки всі тестові випадки виконано, вимоги продуктивності задоволено, всі баги виправлено і виправлення підтверджено, тестування безпеки пройдено.

ВИСНОВКИ

Результатом написання дипломної роботи є готовий програмний продукт, який являє собою веб-орієнтовану систему нарахування стипендії. Ключовою перевагою розробленої системи являється спрощення роботи працівника ВНЗ з процесом нарахування стипендії та супроводжуючих цей процес функцій, без прив'язки до робочого місця з встановленим ПЗ.

Розроблена веб-орієнтована система на відміну від її аналогів, майже повністю автоматизує процес нарахування стипендії шляхом автоматичного визначення рейтингу, можливості отримувати підвищену та пільгову стипендію і допуску до отримання стипендії на основі даних, отриманих із заліково-екзаменаційної сесії.

Окрім цього, за допомогою розробленої системи доступна можливість друку автоматично-згенерованого наказу на призначення стипендії на основі вище-згаданих критеріїв. Також реалізовано механізм сортування студентів по групах та формі навчання, пошук по прізвищу. Ще доступна функція формування звіту із стипендійними даними та реалізовано механізм сортування по групі, наявності пільгової категорії та рейтингу. Реалізована функція обрахунку середнього рейтингу та загальної суми стипендії, відповідно до раніше-згадуваних критеріїв.

Важливою особливістю розробленої веб-орієнтованої системи нарахування стипендії є те, що робота з нею можлива з будь-якого місця, де є доступ до мережі Інтернет.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Депонування коштів [Електронний ресурс]. – Режим доступу: http://leksika.com.ua/17140326/legal/deponuvannya_groshey_koshtiv
2. Стипендія [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D0%B8%D0%BF%D0%B5%D0%BD%D0%B4%D1%96%D1%8F>
3. Індексція [Електронний ресурс]. – Режим доступу: <https://news.dtkr.ua/ua/tags/indeksacija>
4. ПДФО [Електронний ресурс]. – Режим доступу: <http://www.vobu.com.ua/ukr/news/view/50743>
5. Загородній А.Г., Партин Г.О. Бухгалтерський облік: основи теорії та практики: Навч. посіб. — 2-ге вид., перероб. І доп. — К.: Т-во “Знання”, КОО, 2014. —327 с.
6. Горев А., Макашарипов С. «Эффективная работа с СУБД»: СПб, «Питер», 2012. – 512с.
7. Поль Дюбуа: MySQL. Издательство: Вильямс, 2006. —413 с.
8. Астахова И.Ф., Толстобров А.П., Мельников В.М. SQL в примерах и задачах. Воронеж, 2001. 160с.
9. Codd E.F. Further Normalization of the Data base Relational Model //Data Base Systems.- N.J.: Prentice-Hall, 1972. - P.33-64.
10. Чен П. Модель "сущность-связь" - шаг к единому представлению о данных //СУБД. - 2005. - №3. - С.137-158.
11. Мартин Д. Планирование развития автоматизированных систем. - М.: Финансы и статистика, 2009. - 196 с.
12. Кириллов В.В. Структуризованный язык запросов (SQL). - СПб.: ИТМО, 2011. - 80 с.
13. Грабер М. Справочное руководство по SQL. - М.: Лори, 1997. - 291 с.
14. Кузнецов С. Д. Основы баз данных. – 2-е изд. –М.: Интернет-университет информационных технологий; БИНОМ. Лаборатория знаний, 2007.

15. 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, 2004. - 416с.
16. Тиори Т., Фрай Дж. Проектирование структур баз данных. Т.1,2, - М.: Мир,1985. -287с., -320с.
17. Майерс Г. Искусство тестирования программ / Пер. с англ. под ред. Б. А. Позина. – М.: Финансы и статистика, 2009. – 176 с.
18. Myers G. J. A Controlled Experiment in Programm Testing and Code Walk-throughs/Inspections. – Commun. ACM, 1978, 21(9), p. 760–768.
19. Apache HTTP Server. [Электронный ресурс]. – Режим доступа: https://uk.wikipedia.org/wiki/Apache_HTTP_Server
20. Как разместить сайт в Интернет. Хостинг и домен для сайта. [Электронный ресурс]. – Режим доступа: <http://bitte.net.ua/blog/kak-zalit-sayt-na-hosting/>
21. Загрузка сайта на хостинг. [Электронный ресурс]. – Режим доступа: <https://myrusakov.ru/site-to-hosting.html>

ДОДАТОК А

ВИХІДНИЙ КОД РОЗРОБКИ ВЕБ-ОРІЄНТОВАНОЇ СИСТЕМИ НАРАХУВАННЯ СТИПЕНДІЇ

Index.php

```

<?php
session_start();
echo<link rel="stylesheet" href="../css/style.css"/>
<script type="text/javascript" src="js/jquery.1.7.1.min.js"></script>
<script type="text/javascript" src="js/script1.js"></script>;
class SITE{
function go(){
$connect = mysql_connect('localhost','root','') or die(mysql_error());
mysql_select_db('dyplom');
if (isset($_POST['enter'])){
$_login = $_POST['e_login'];
$_password = md5($_POST['e_password']);
$query = mysql_query("SELECT * FROM users WHERE login='$_login'");
$user_data = mysql_fetch_array($query);
if ($user_data['password']== $_password) {
$_SESSION['usr'] = $_login;
$_SESSION['position'] = $user_data['position'];
}
else {
echo "<h2 align='center' style='color:#383838 ;background:#C3BEBA; font-family: Lobster,
cursive;'>Помилка вводу логіна/пароля!</h2>";
}
}
if(isset($_POST['logout'])){
unset($_SESSION['usr']);
session_destroy();
}
$b = new LOADER();
if (isset($_SESSION['usr']))
{
echo "<title>Стипендія</title>";
$b -> showBlocks();
}
}

```

```

$b -> loadScript();
}
else
{
$b -> authBuild();
}
}
}
class LOADER{
function loadScript(){
echo"<script>
function showContent(link) {
var cont = document.getElementById('contentBody');
var loading = document.getElementById('loading');
cont.innerHTML = loading.innerHTML;
var http = createRequestObject();
if( http ) {
http.open('get', link);
http.onreadystatechange = function () {
if(http.readyState == 4) {
cont.innerHTML = http.responseText;
}
}
http.send(null);
} else {
document.location = link;
}
}
function createRequestObject() {
try { return new XMLHttpRequest() }
catch(e) {
try { return new ActiveXObject('Msxml2.XMLHTTP') }
catch(e) {
try { return new ActiveXObject('Microsoft.XMLHTTP') }
catch(e) { return null; }
}
}
}
}
</script>";

```

```

}
function showBlocks(){
echo<table id="entered" style="top: 145px;">;
$this -> menuBuild();
$this -> contentBuild();
$this -> footerBuild();
echo</table>;
echo<div id="loading" style="display: none">;
echo;
echo</div>;
}
function authBuild(){
echo<div id="login_container">;
echo<div id="form_container">;
echo<p class="login-text">Авторизація на сайті</p>;
echo<form id="enter" method="POST" action="index.php">;
echo<input type="text" id="e_log" name="e_login" class="text_input" placeholder=" Логін"
required/><br>;
echo<input type="password" id="e_pass" class="text_input" name="e_password" placeholder="
Пароль" required /><br>;
echo<input type="submit" id="e_button" value="Вхід" name="enter" />;
echo</form>;
echo</div>;
echo</div> ';
}
function menuBuild(){
echo "<div id='header'>
<nav id='menu-wrap'>
<ul id='menu'>
<form align = 'right'; id='a_log' method='post' action='index.php'>
<li><a href='#' onclick=showContent('forms/headerBuild.php');>Головна</a></li>
<li>
<a href="#">Студенти</a>
<ul>
<li>
<a href='#' onclick=showContent('action/students/view_students_form_of_education_all.php');>Відкрити
список</a>
</li>
<li>

```

```

<a href="#" onclick=showContent('action/students/add_student.php')>Додати нового</a>
</li>
<li>
<a href="#" onclick=showContent('action/students/update_students.php')>Редагувати</a>
</li>
<li>
<a href="#" onclick=showContent('action/students/delete_student.php')>Видалити</a>
</li>
</ul>
</li>
<li>
<a href="#">Стипендія</a>
<ul>
<li>
<a href="#" onclick=showContent('action/stypendia/gpaRating_and_bursarySum.php')>Нарахування</a>
</li>";
if ($_SESSION['position'] !== 'Бухгалтер')    {
echo "<li>
<a href="#">Редагувати</a>
<ul>
<li>
<a href="#" onclick=showContent('action/stypendia/update_date.php')>Дату дії</a>
</li>
<a href="#" onclick=showContent('action/stypendia/update_st_type.php')>Суму підвищеної ст.</a>
</li>
<li>
<a href="#" onclick=showContent('action/stypendia/update_baz_st.php')>Базову стипендію</a>
</li>
</ul>
</li>
<li>";
}
echo"          </li>
</ul>
</li>
<li>
<a href="#">Звіти</a>
<ul>
<li>

```

```

<a href='action/report/showReport.php'>Сформувати наказ</a>
</li>
<li>
<a href='action/report/view_report_search_result.php'>Перегляд</a>
</li>
</ul>
";
if ($_SESSION['position'] == 'Адмін'){
echo
"
<li>
<a href="">Користувачі</a>
<ul>
<li>
<a href='#' onclick=showContent('action/users/view_users.php');>Перегляд</a>
</li>
<li>
<a href='#' onclick=showContent('action/users/update_users.php')> Редагувати </a>
</li>
<li>
<a href='#' onclick=showContent('action/users/add_user.php')>Додати нового</a>
</li>
<li>
<a href='#' onclick=showContent('action/users/delete_user.php')>Видалити</a>
</li>
</ul>
</li>
<input id='exit_btn' type='submit' name='logout' value="" >";
}
else {
echo "<right><input id='exit_btn' type='submit' name='logout' value="" style='margin-left:
445px;'/></right>";}
echo"
</form>
</ul>
</nav>
</td>
</div>
";

```

```

}
function contentBuild(){
echo'
<div>
<td align="center">
<div id="contentBody">
</div>
</td>
</div>
';
}
function footerBuild(){
echo'
<tr>
<td id=footer align="center">
<hr color="#373737">
<h5>© 2016 | <a id= "author" href="http://vk.com/just1_4fun" target="_blank" > Дубчак Руслан</a></h5>
</td>
</tr>
';
}
}
$openSite = new SITE();
$openSite -> go();
?>

```

Stypendia.php

```

<?php
class STYPENDIA{
function ConnectToDB(){
$hostname = 'localhost';
$username = 'root';
$password = '';
$dbName = "dyplom";
mysql_connect($hostname, $username, $password) or die ("Не можливо створити з'єднання!");
mysql_select_db($dbName) or die (mysql_error());

```

```

}
function
CalculateGPA_and_bursarySum($load_data_query,$max_students_id_query,$update_GPA_query){
$max_students_id_query =(" SELECT MAX(students_id) as m_s_id FROM stypendia ");
$res = mysql_query($load_data_query) or die(mysql_error());
$max_student_id = mysql_query($max_students_id_query) or die(mysql_error());
while ($m_s_id_arr = mysql_fetch_array($max_student_id)) {
echo "<tr>\n";
$m_s_id = $m_s_id_arr['m_s_id'];
for ($i=1; $i<=$m_s_id;$i++){
$update_GPA_query =("UPDATE stypendia
SET gpa_rating = (
Select avg(mark.rating_fifth_s)
FROM mark
INNER JOIN `session` ON `session`.mark_id = mark.id
WHERE `session`.student_id = '$i'
)
WHERE stypendia.students_id = '$i'
");
$res3 = mysql_query($update_GPA_query) or die(mysql_error());
}
}
echo (
<table class="simple-little-table" border="1" cellspacing="0">
<tr style=\\"border: solid 1px #000\\">
<td align="center"><b>Група</b></td>
<td align="center"><b>Прізвище</b></td>
<td align="center"><b>Ім*я</b></td>
<td align="center"><b>По батькові</b></td>
<td align="center"><b>Пільгова категорія</b></td>
<td align="center"><b>Середній бал</b></td>
<td align="center"><b>Допуск</b></td>
<td align="center"><b>Сума, грн</b></td>
<a id ="1">
</tr>
);
while ($row = mysql_fetch_array($res)) {
echo "<tr>\n";
echo "<td>".$row['grup'].</td>\n";

```



```

echo "<td>".$row['surname']."</td>\n";
echo "<td>".$row['student_name']."</td>\n";
echo "<td>".$row['f_name']."</td>\n";
echo "<td>".$row['preferential_category']."</td>\n";
echo "<td>".$row['gpa_rating']."</td>\n";
echo "<td>".$row['admission']."</td>\n";
echo "<td>".$row['bursary_sum']."</td>\n";
}
echo ("</table>\n </a>");
$res = mysql_query('CALL `calculate_stypendia`') or die(mysql_error());
$max_student_id2 = mysql_query($max_students_id_query) or die(mysql_error());
while ($m_s_id_arr2 = mysql_fetch_array($max_student_id2)) {
$m_s_id2 = $m_s_id_arr2['m_s_id'];
for ($j=1; $j<=$m_s_id2;$j++){
$update_bursary_sum_query =("UPDATE stypendia
SET bursary_sum = (
SELECT `sum` FROM (
    SELECT suma + pref_cat_sum + add_sum as `sum`
    FROM stypendia
    INNER JOIN baz_st on baz_st.id = stypendia.baz_st_id
    INNER JOIN preferential_category on preferential_category.id = stypendia.preferential_category_id
    INNER JOIN st_type on st_type.id = stypendia.st_type_id
    WHERE stypendia.students_id = '$j'
    )
AS ResultSet
)
WHERE stypendia.students_id = '$j' AND stypendia.admission = 'yes'
");
$resultat = mysql_query($update_bursary_sum_query) or die(mysql_error());
}
}
mysql_close();
}
function updateDate(){
if(@$_POST['edit'])
{
$query = "UPDATE date SET date1='{$_POST['edit_date1']}', date2='{$_POST['edit_date2']}'
WHERE id='{$_POST['update']}'";
mysql_query($query) or die (mysql_error());
}
}

```

```

die (require_once "../forms/well_done.php");
}
$query = "SELECT * FROM date";
$res = mysql_query($query) or die(mysql_error());
$row = mysql_num_rows($res);
while ($row = mysql_fetch_array($res))
{
echo "<form action='action/stypendia/update_date.php' method='post' name='edit_form'>\n";
echo "<input type='hidden' name='update' value='".$row["id"]."' />\n";
echo "<table class='simple-little-table' border='1' cellspacing='0'>\n";
echo "<tr>\n";
echo "    <td colspan='2' style='border-bottom:solid 1px #CCCCCC;'><b><i><div
id='num'>#".$row["id"]."</div></b></i></td>\n";
echo "</tr><tr>\n";
echo "    <td>Дата початку дії стипендії:</td><td><input type='text' value='".$row['date1']."'
name='edit_date1' /></td>\n";
echo "</tr><tr>\n";
echo "    <td>Дата кінця дії стипендії:</td><td><input type='text' value='".$row['date2']."'
name='edit_date2' /></td>\n";
echo "</tr><tr>\n";
echo "    <td colspan='2' align='center'><input type='submit' name='edit' class='buttons'
value='Зберегти зміни' /></td>\n";
echo "</tr></table></form>\n\n";
}
echo"</div>";
mysql_close();
}
function update_st_type(){
if(@$_POST['edit'])
{
$query = "UPDATE st_type SET add_sum='{$_POST['edit_add_sum']}'
WHERE id='{$_POST['update']}'";
mysql_query($query) or die (mysql_error());
die (require_once "../forms/well_done.php");
}
$query = "SELECT * FROM st_type";
$res = mysql_query($query) or die(mysql_error());
$row = mysql_num_rows($res);

```

```

while ($row = mysql_fetch_array($res))
{
echo "<form action=\"action/stypendia/update_st_type.php\" method=\"post\" name=\"edit_form\">\n";
echo "<input type=\"hidden\" name=\"update\" value=\"\".$row[\"id\"].\" \" />\n";
echo "<table class=\"simple-little-table\" border=\"1\" cellspacing=\"0\">\n";
echo "<tr>\n";
echo "    <td colspan=\"2\" style=\"border-bottom:solid 1px #CCCCCC;\"><b><i><div
id=\"num\">#\".$row[\"id\"].\"</div></b></i></td>\n";
echo "</tr><tr>\n";
echo "    <td>Тип стипендії:</td><td><input type=\"text\" disabled value=\"\".$row[\"st_type_name\"].\" \"
name=\"edit_st_type_name\" /></td>\n";
echo "</tr><tr>\n";
echo "    <td>Сума надбавки:</td><td><input type=\"text\" value=\"\".$row[\"add_sum\"].\" \"
name=\"edit_add_sum\" /></td>\n";
echo "</tr><tr>\n";
echo "    <td colspan=\"2\" align=\"center\"><input type=\"submit\" name=\"edit\" class=\"buttons\"
value=\"Зберегти зміни\" /></td>\n";
echo "</tr></table></form>\n\n";
}
echo"</div>";
mysql_close();
}
function update_baz_st(){
if(@$_POST['edit'])
{
$query = "UPDATE baz_st SET suma='{$_POST['edit_suma']}'
WHERE id='{$_POST['update']}'";
mysql_query($query) or die (mysql_error());
die (require_once "../forms/well_done.php");
}
$query = "SELECT * FROM baz_st";
$res = mysql_query($query) or die(mysql_error());
$row = mysql_num_rows($res);
while ($row = mysql_fetch_array($res))
{
echo "<form action=\"action/stypendia/update_baz_st.php\" method=\"post\" name=\"edit_form\">\n";
echo "<input type=\"hidden\" name=\"update\" value=\"\".$row[\"id\"].\" \" />\n";
echo "<table class=\"simple-little-table\" border=\"1\" cellspacing=\"0\">\n";
echo "<tr>\n";

```

```

echo "<td colspan=\"2\" style=\"border-bottom:solid 1px #CCCCCC;\"><b><i><div
id=\"num\">#\".$row[\"id\"].\"</div></b></i></td>\n";
echo "</tr><tr>\n";
echo "<td>Освітній рівень:</td><td><input type=\"text\" disabled value=\"\".$row[\"ed_level\"].\"
name=\"edit_ed_level\" /></td>\n";
echo "</tr><tr>\n";
echo "<td>Сума стипендії:</td><td><input type=\"text\" value=\"\".$row[\"suma\"].\" name=\"edit_suma\"
/></td>\n";
echo "</tr><tr>\n";
echo "<td colspan=\"2\" align=\"center\"><input type=\"submit\" name=\"edit\" class=\"buttons\"
value=\"Зберегти зміни\" /></td>\n";
echo "</tr></table></form>\n\n";
}
echo"</div>";
mysql_close();
}
}
?>

```

ДОДАТОК Б

РЕЗУЛЬТАТ ФІЗИЧНОГО ПРОЕКТУВАННЯ БД

```

-- phpMyAdmin SQL Dump
-- version 4.0.10.10
-- http://www.phpmyadmin.net
--
-- Хост: 127.0.0.1:3306
-- Время создания: Май 29 2016 г., 23:54
-- Версия сервера: 5.5.45
-- Версия PHP: 5.3.29
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
--
-- База данных: `diplom`
--
DELIMITER $$
--
-- Процедуры
--
CREATE DEFINER=`root`@`%` PROCEDURE `calculate_stypendia`()
    MODIFIES SQL DATA
    BEGIN

        UPDATE stypendia
        SET stypendia.st_type_id = (SELECT id from st_type where st_type.st_type_name
        ='Звичайна')
        WHERE stypendia.gpa_rating >=4 and stypendia.gpa_rating<5;

        UPDATE stypendia
        SET stypendia.st_type_id = (SELECT id from st_type where st_type.st_type_name
        ='Підвищена')
        WHERE stypendia.gpa_rating =5;

        UPDATE stypendia
        SET stypendia.st_type_id = (SELECT id from st_type where st_type.st_type_name
        ='Звичайна')
        WHERE stypendia.gpa_rating >=3 and stypendia.gpa_rating<4 and
        stypendia.preferential_category_id != (SELECT id from preferential_category WHERE
        preferential_category.name='Немає');

        UPDATE stypendia
        SET stypendia.admission = 'no'
        WHERE stypendia.gpa_rating <4;

        UPDATE stypendia
        SET stypendia.admission = 'yes'
        WHERE stypendia.gpa_rating >=4;
    
```

```

UPDATE stypendia
SET stypendia.admission = 'yes'
WHERE stypendia.preferential_category_id != (SELECT id from preferential_category
WHERE preferential_category.name='Немає');

```

```

UPDATE stypendia
SET stypendia.admission = 'no'
WHERE stypendia.gpa_rating <3;

```

```

UPDATE stypendia
SET stypendia.bursary_sum = 0
WHERE stypendia.admission ='no';

```

```
END$$
```

```
DELIMITER ;
```

```
-----
```

```
--
-- Структура таблицы `baz_st`
--
```

```

CREATE TABLE IF NOT EXISTS `baz_st` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `ed_level` varchar(50) NOT NULL,
  `suma` double NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=5 ;

```

```
--
-- Дамп данных таблицы `baz_st`
--
```

```

INSERT INTO `baz_st` (`id`, `ed_level`, `suma`) VALUES
(1, 'Магістр', 916.56),
(2, 'Спеціаліст', 900.15),
(3, 'Бакалавр', 816.44),
(4, 'Молодший спеціаліст', 544.5);

```

```
-----
```

```
--
-- Структура таблицы `date`
--
```

```

CREATE TABLE IF NOT EXISTS `date` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `date1` date NOT NULL,
  `date2` date NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=2 ;

```

```
--
-- Дамп данных таблицы `date`
--
```

```
INSERT INTO `date` (`id`, `date1`, `date2`) VALUES
(1, '2016-05-08', '2016-12-08');

-----

--
-- Структура таблицы `form_of_education`
--

CREATE TABLE IF NOT EXISTS `form_of_education` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(50) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=3 ;

--
-- Дамп данных таблицы `form_of_education`
--

INSERT INTO `form_of_education` (`id`, `name`) VALUES
(1, 'Державник'),
(2, 'Платник');

-----

--
-- Структура таблицы `grup`
--

CREATE TABLE IF NOT EXISTS `grup` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(50) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=11 ;

--
-- Дамп данных таблицы `grup`
--

INSERT INTO `grup` (`id`, `name`) VALUES
(1, 'pzs-11'),
(2, 'pzs-12'),
(3, 'pzs-21'),
(4, 'pzs-22'),
(5, 'pzs-31'),
(6, 'pzs-32'),
(7, 'pzs-41'),
(8, 'pzs-42'),
(9, 'pzs-51'),
(10, 'pzs-52');

-----

--
-- Структура таблицы `mark`
--
```

```

CREATE TABLE IF NOT EXISTS `mark` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `rating` varchar(50) NOT NULL,
  `rating_bol_min` int(11) NOT NULL,
  `rating_bol_max` int(11) NOT NULL,
  `rating_fifth_s` float NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=8 ;

--
-- Дамп данных таблицы `mark`
--

INSERT INTO `mark` (`id`, `rating`, `rating_bol_min`, `rating_bol_max`, `rating_fifth_s`)
VALUES
(1, 'A', 90, 100, 5),
(2, 'B', 85, 89, 4),
(3, 'C', 75, 84, 4),
(4, 'D', 70, 74, 3),
(5, 'E', 60, 69, 3),
(6, 'FX', 35, 59, 2),
(7, 'F', 0, 34, 2);

-----

--
-- Структура таблицы `personal_data`
--

CREATE TABLE IF NOT EXISTS `personal_data` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `surname` varchar(50) CHARACTER SET cp1251 NOT NULL,
  `name` varchar(50) CHARACTER SET cp1251 NOT NULL,
  `f_name` varchar(50) CHARACTER SET cp1251 NOT NULL,
  `id_number` int(11) NOT NULL,
  `form_of_education_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `form_of_education_id` (`form_of_education_id`)
) ENGINE=InnoDB DEFAULT CHARSET=cp1251 COLLATE=cp1251_ukrainian_ci
AUTO_INCREMENT=42 ;

--
-- Дамп данных таблицы `personal_data`
--

INSERT INTO `personal_data` (`id`, `surname`, `name`, `f_name`, `id_number`,
`form_of_education_id`) VALUES
(1, 'Назаренко', 'Назар', 'Назарович', 666666666, 2),
(2, 'Мартиненко', 'Григорій', 'Йосипович', 555555555, 1),
(3, 'Андруссяк', 'Іван', 'Степанович', 111111111, 1),
(4, 'Ігоренко', 'Петро', 'Андрійович', 333333333, 1),
(5, 'Потоцький', 'Юрій', 'Сергійович', 777777777, 2),
(40, 'Ковальчук', 'Іван', 'Іванович', 444444444, 1),
(41, 'Бойко', 'Микола', 'Ігорович', 222222222, 1);

-----

--

```



```
-- Структура таблицы `preferential_category`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `preferential_category` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(50) NOT NULL,
  `pref_cat_sum` float NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=5 ;
```

```
--
```

```
-- Дамп данных таблицы `preferential_category`
```

```
--
```

```
INSERT INTO `preferential_category` (`id`, `name`, `pref_cat_sum`) VALUES
(1, 'Немає', 0),
(2, 'Сироти', 600),
(3, 'Батьки молодих сімей', 700),
(4, 'Жителі гірських місцевостей', 800);
```

```
-----
```

```
--
```

```
-- Структура таблицы `session`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `session` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `student_id` int(11) NOT NULL,
  `subject_id` int(11) NOT NULL,
  `teacher_id` int(11) NOT NULL,
  `mark_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `student_id` (`student_id`),
  KEY `subject_id` (`subject_id`),
  KEY `teacher_id` (`teacher_id`),
  KEY `mark_id` (`mark_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=31 ;
```

```
--
```

```
-- Дамп данных таблицы `session`
```

```
--
```

```
INSERT INTO `session` (`id`, `student_id`, `subject_id`, `teacher_id`, `mark_id`) VALUES
(1, 1, 1, 4, 1),
(2, 1, 2, 5, 1),
(3, 1, 3, 2, 1),
(4, 1, 4, 3, 1),
(5, 1, 5, 1, 1),
(6, 1, 6, 5, 1),
(7, 2, 6, 5, 3),
(8, 2, 6, 5, 3),
(9, 2, 6, 5, 3),
(10, 2, 6, 5, 3),
(11, 2, 6, 5, 3),
(12, 2, 6, 5, 3),
(13, 3, 6, 5, 4),
(14, 3, 6, 5, 4),
```

```
(15, 3, 6, 5, 4),
(16, 3, 6, 5, 3),
(17, 3, 6, 5, 3),
(18, 3, 6, 5, 3),
(19, 37, 6, 5, 3),
(20, 37, 6, 5, 3),
(21, 37, 6, 5, 3),
(22, 37, 6, 5, 1),
(23, 37, 6, 5, 1),
(24, 37, 6, 5, 1),
(25, 38, 6, 5, 1),
(26, 38, 6, 5, 1),
(27, 38, 6, 5, 1),
(28, 38, 6, 5, 1),
(29, 38, 6, 5, 1),
(30, 38, 6, 5, 1);
```

```
-----
```

```
--
-- Структура таблицы `students`
--
```

```
CREATE TABLE IF NOT EXISTS `students` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `grup_id` int(11) NOT NULL,
  `personal_data_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `personal_data_id` (`personal_data_id`),
  KEY `grup_id` (`grup_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=39 ;
```

```
--
-- Дамп данных таблицы `students`
--
```

```
INSERT INTO `students` (`id`, `grup_id`, `personal_data_id`) VALUES
(1, 7, 3),
(2, 7, 4),
(3, 7, 2),
(4, 7, 1),
(8, 7, 5),
(37, 7, 40),
(38, 7, 41);
```

```
-----
```

```
--
-- Структура таблицы `stypendia`
--
```

```
CREATE TABLE IF NOT EXISTS `stypendia` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `students_id` int(11) DEFAULT NULL,
  `date_id` int(11) DEFAULT NULL,
  `baz_st_id` int(11) DEFAULT NULL,
  `preferential_category_id` int(11) DEFAULT NULL,
  `st_type_id` int(11) DEFAULT NULL,
```

```

`gpa_rating` float DEFAULT NULL,
`admission` varchar(4) DEFAULT NULL,
`bursary_sum` double DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `date_id` (`date_id`),
KEY `baz_st_id` (`baz_st_id`),
KEY `st_type_id` (`st_type_id`),
KEY `preferential_category_id` (`preferential_category_id`),
KEY `students_id` (`students_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=19 ;

--
-- Дамп данных таблицы `stypendia`
--

INSERT INTO `stypendia` (`id`, `students_id`, `date_id`, `baz_st_id`, `preferential_category_id`,
`st_type_id`, `gpa_rating`, `admission`, `bursary_sum`) VALUES
(1, 1, 1, 3, 2, 2, 5, 'yes', 1516.44),
(2, 2, 1, 2, 4, 1, 4, 'yes', 1700.15),
(5, 3, 1, 4, 1, 1, 3.5, 'no', 0),
(17, 37, 1, 3, 1, 1, 4.5, 'yes', 816.44),
(18, 38, 1, 3, 1, 2, 5, 'yes', 916.44);

-----

--
-- Структура таблицы `st_type`
--

CREATE TABLE IF NOT EXISTS `st_type` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `add_sum` float NOT NULL,
  `st_type_name` varchar(50) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=3 ;

--
-- Дамп данных таблицы `st_type`
--

INSERT INTO `st_type` (`id`, `add_sum`, `st_type_name`) VALUES
(1, 0, 'Звичайна'),
(2, 100, 'Підвищена');

-----

--
-- Структура таблицы `subject`
--

CREATE TABLE IF NOT EXISTS `subject` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(50) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=9 ;

--
-- Дамп данных таблицы `subject`

```

```

--

INSERT INTO `subject` (`id`, `name`) VALUES
(1, 'Геометрія'),
(2, 'Алгебра'),
(3, 'Тестування ПЗ'),
(4, 'Мікропроцесорні системи'),
(5, 'Основи охорони праці'),
(6, 'Групова динаміка і комунікації'),
(7, 'Проектування БД'),
(8, 'Безпека програм та даних');

-----

--
-- Структура таблиці `teacher`
--

CREATE TABLE IF NOT EXISTS `teacher` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(50) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=6 ;

--
-- Дамп даних таблиці `teacher`
--

INSERT INTO `teacher` (`id`, `name`) VALUES
(1, 'Костенко І. І.'),
(2, 'Тарасенко М. Ю.'),
(3, 'Миколаєв В. Г.'),
(4, 'Панда І. М.'),
(5, 'Поторчук Л. Н.');
```

```

-----

--
-- Структура таблиці `users`
--

CREATE TABLE IF NOT EXISTS `users` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `login` varchar(50) NOT NULL,
  `password` varchar(50) NOT NULL,
  `position` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=9 ;

--
-- Дамп даних таблиці `users`
--

INSERT INTO `users` (`id`, `login`, `password`, `position`) VALUES
(1, 'admin', '21232f297a57a5a743894a0e4a801fc3', 'Адмін'),
(2, 'user1', '24c9e15e52afc47c225b757e7bee1f9d', 'Головний бухгалтер'),
(7, 'user2', '7e58d63b60197ceb55a1c487989a3720', 'Бухгалтер'),
(8, 'л', '1a41a6326d10d33048737d57b9fa866c', 'Адмін');
```

```

--
-- Ограничения внешнего ключа сохраненных таблиц
--

--
-- Ограничения внешнего ключа таблицы `personal_data`
--
ALTER TABLE `personal_data`
  ADD CONSTRAINT `form_of_education_id` FOREIGN KEY (`form_of_education_id`)
REFERENCES `form_of_education` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION;

--
-- Ограничения внешнего ключа таблицы `session`
--
ALTER TABLE `session`
  ADD CONSTRAINT `mark_id` FOREIGN KEY (`mark_id`) REFERENCES `mark` (`id`) ON
DELETE NO ACTION ON UPDATE NO ACTION,
  ADD CONSTRAINT `student_id` FOREIGN KEY (`student_id`) REFERENCES `students` (`id`)
ON DELETE NO ACTION ON UPDATE NO ACTION,
  ADD CONSTRAINT `subject_id` FOREIGN KEY (`subject_id`) REFERENCES `subject` (`id`)
ON DELETE NO ACTION ON UPDATE NO ACTION,
  ADD CONSTRAINT `teacher_id` FOREIGN KEY (`teacher_id`) REFERENCES `teacher` (`id`)
ON DELETE NO ACTION ON UPDATE NO ACTION;

--
-- Ограничения внешнего ключа таблицы `students`
--
ALTER TABLE `students`
  ADD CONSTRAINT `grup_id` FOREIGN KEY (`grup_id`) REFERENCES `grup` (`id`) ON
DELETE NO ACTION ON UPDATE NO ACTION,
  ADD CONSTRAINT `personal_data_id` FOREIGN KEY (`personal_data_id`) REFERENCES
`personal_data` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Ограничения внешнего ключа таблицы `stypendia`
--
ALTER TABLE `stypendia`
  ADD CONSTRAINT `baz_st_id` FOREIGN KEY (`baz_st_id`) REFERENCES `baz_st` (`id`) ON
DELETE NO ACTION ON UPDATE NO ACTION,
  ADD CONSTRAINT `date_id` FOREIGN KEY (`date_id`) REFERENCES `date` (`id`) ON
DELETE NO ACTION ON UPDATE NO ACTION,
  ADD CONSTRAINT `preferential_category_id` FOREIGN KEY (`preferential_category_id`)
REFERENCES `preferential_category` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  ADD CONSTRAINT `students_id` FOREIGN KEY (`students_id`) REFERENCES `students`
(`id`) ON DELETE CASCADE ON UPDATE NO ACTION,
  ADD CONSTRAINT `st_type_id` FOREIGN KEY (`st_type_id`) REFERENCES `st_type` (`id`)
ON DELETE NO ACTION ON UPDATE NO ACTION;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

ДОДАТОК В

РЕЗУЛЬТАТИ ТЕСТУВАННЯ

Test suite

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <meta content="text/html; charset=UTF-8" http-equiv="content-type" />
  <title>Test Suite</title>
</head>
<body>
<table id="suiteTable" cellpadding="1" cellspacing="1" border="1" class="selenium"><tbody>
<tr><td><b>Test Suite</b></td></tr>
<tr><td><a href="authorization_test.html">Authorization_test</a></td></tr>
<tr><td><a href="all_tests.html">Add_new_student_test</a></td></tr>
<tr><td><a href="all_tests.html">Search_new_student_test</a></td></tr>
<tr><td><a href="1.html">Update_student_test</a></td></tr>
<tr><td><a href="2.html">Delete_student_test</a></td></tr>
</tbody></table>
</body>
</html>
```

Add_new_student_test_case

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://selenium-ide.openqa.org/profiles/test-case">
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <link rel="selenium.base" href="http://diplom.loc/" />
  <title>Add_new_student_test</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">Add_new_student_test</td></tr>
</thead><tbody>
<tr>
  <td>open</td>
  <td></td>
  <td></td>
</tr>
<tr>
  <td>click</td>
  <td>link=Додати нового</td>
  <td></td>
</tr>
<tr>
  <td>type</td>
  <td>name=surname</td>
  <td>Дубчак</td>
</tr>
```

```

</tr>
<tr>
  <td>type</td>
  <td>name=name</td>
  <td>Руслан</td>
</tr>
<tr>
  <td>type</td>
  <td>name=f_name</td>
  <td>ИВАНОВИЧ</td>
</tr>
<tr>
  <td>type</td>
  <td>name=id_number</td>
  <td>123456789</td>
</tr>
<tr>
  <td>select</td>
  <td>name=grup_id</td>
  <td>label=гзс-41</td>
</tr>
<tr>
  <td>select</td>
  <td>name=baz_st_id</td>
  <td>label=Бакалавр</td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>name=reg_btn</td>
  <td></td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>link=OK</td>
  <td></td>
</tr>
</tbody></table>
</body>
</html>

```

Authorization_test_case

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://selenium-ide.openqa.org/profiles/test-case">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="http://diplom.loc/" />
<title>Autorization_test</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">Authorization_test</td></tr>
</thead><tbody>

```

```

<tr>
  <td>open</td>
  <td>/</td>
  <td></td>
</tr>
<tr>
  <td>type</td>
  <td>id=e_log</td>
  <td>admin</td>
</tr>
<tr>
  <td>type</td>
  <td>id=e_pass</td>
  <td>admin</td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>id=e_button</td>
  <td></td>
</tr>
</tbody></table>
</body>
</html>

```

Delete_student_test_case

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://selenium-ide.openqa.org/profiles/test-case">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="http://diplom.loc/" />
<title>Delete_student_test</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">Delete_student_test</td></tr>
</thead><tbody>
<tr>
  <td>open</td>
  <td>/action/students/view_students_search_result.php?Search_select_form_of_education=%25%D0%
  %B0%25&amp;searchSurname=%D0%B4%D1%83%D0%B1%D1%87&amp;search_b=&amp;Search_sele
  ct_grup=%25-%25</td>
  <td></td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>id=back_b</td>
  <td></td>
</tr>
<tr>
  <td>click</td>
  <td>link=Видалити</td>
  <td></td>
</tr>

```



```

<tr>
  <td>clickAndWait</td>
  <td>xpath=("//a[contains(text(),'Видалити')])[10]</td>
  <td></td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>css=a &gt; img</td>
  <td></td>
</tr>
<tr>
  <td>click</td>
  <td>link=Відкрити список</td>
  <td></td>
</tr>
<tr>
  <td>type</td>
  <td>id=searchSurname</td>
  <td>Дубч</td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>id=search_b</td>
  <td></td>
</tr>
</tbody></table>
</body>
</html>

```

Search_new_student_test_case

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://selenium-ide.openqa.org/profiles/test-case">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="http://dyplom.loc/" />
<title>Search_new_student_test</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">Search_new_student_test</td></tr>
</thead><tbody>
<tr>
  <td>open</td>
  <td>/index.php</td>
  <td></td>
</tr>
<tr>
  <td>click</td>
  <td>link=Відкрити список</td>
  <td></td>
</tr>
</tbody>
</table>

```

```

        <td>type</td>
        <td>id=searchSurname</td>
        <td>Дубч</td>
</tr>
<tr>
        <td>clickAndWait</td>
        <td>id=search_b</td>
        <td></td>
</tr>

</tbody></table>
</body>
</html>

```

Update_student_test_case

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://selenium-ide.openqa.org/profiles/test-case">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="http://diplom.loc/" />
<title>Update_student_test</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">Update_student_test</td></tr>
</thead><tbody>
<tr>
        <td>open</td>
        <td>/action/students/view_students_search_result.php?Search_select_form_of_education=%25%D0
%B0%25&amp;searchSurname=%D0%94%D1%83%D0%B1%D1%87&amp;search_b=&amp;Search_sele
ct_grup=%25-%25</td>
        <td></td>
</tr>
<tr>
        <td>clickAndWait</td>
        <td>id=back_b</td>
        <td></td>
</tr>
<tr>
        <td>click</td>
        <td>link=Редагувати</td>
        <td></td>
</tr>
<tr>
        <td>select</td>
        <td>xpath=(//select[@name='edit_form_of_education_id'])[8]</td>
        <td>label=Платник</td>
</tr>
<tr>
        <td>clickAndWait</td>
        <td>xpath=(//input[@name='edit'])[8]</td>
        <td></td>
</tr>

```

```
<tr>
  <td>clickAndWait</td>
  <td>css=a &gt; img</td>
  <td></td>
</tr>
<tr>
  <td>click</td>
  <td>link=Відкрити список</td>
  <td></td>
</tr>
<tr>
  <td>type</td>
  <td>id=searchSurname</td>
  <td>дубч</td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>id=search_b</td>
  <td></td>
</tr>
</tbody></table>
</body>
</html>
```

ДОДАТОК Г

Таблиця 1

Функціональні тестові випадки

Id	Модуль	Назва	Кроки	Очікуваний результат	Статус
1	Auth	Авторизація	1. Перехід на сторінку авторизації 2. Заповнення форми 3. Натискання кнопки “вхід”	Відкриття основного функціоналу системи	Виконано
2	Student	Перегляд списку студентів	1. Вибір пункту меню: студенти -> відкрити список	сформований список студентів та їх даних	Виконано
3	Student	Пошук студента	1. Вибір пункту меню: студенти -> відкрити список 2. Введення пошукового запиту 3. Натискання кнопки “пошук”	Перенаправлення на сторінку із результатами пошуку	Виконано
4	Student	Додавання студента	1. Вибір пункту меню: студенти -> додати нового 2. Заповнення форми 3. Натискання кнопки “зберегти”	Успішно додані дані студента у БД	Виконано
5	Student	Редагування даних студента	1. Вибір пункту меню: студенти -> редагувати 2. Внесення змін 3. Натискання кнопки “зберегти”	Успішно відредаговані дані студента у БД	Виконано

Продовження таблиці 1

6	Student	Видалення студента	<ol style="list-style-type: none"> 1. Вибір пункту меню: студенти -> видалити 2. Вибір студента 3. Натискання кнопки “видалити” 	Успішно видалені дані студента із БД	Виконано
7	Styp	Нарахування	<ol style="list-style-type: none"> 1. Вибір пункту меню: стипендія -> нарахування 	Успішно обрахований середній бал та сума стипендії, присвоєння дозволу для отримання стипендії і відображення цих відомостей	Виконано
8	Styp	Редагування базових відомостей про стипендію	<ol style="list-style-type: none"> 5. Вибір пункту меню: стипендія -> редагувати 6. Вибір критерію редагування 7. Внесення змін 8. Натискання кнопки “зберегти” 	Успішно відредаговані базові відомості про стипендію у БД	Виконано
9	Report	Сформувати наказ	<ol style="list-style-type: none"> 1. Вибір пункту меню: звіт -> сформувати наказ 	Сформований для друку наказ для отримання стипендії	Виконано
10	Report	Перегляд звіту по отримуваній стипендії	<ol style="list-style-type: none"> 1. Вибір пункту меню: звіти -> перегляд 2. Вибір критеріїв формування звіту 3. Натискання кнопки “пошук” 	Перенаправлення на сторінку із сформованим списком студентів та їх стипендійних даних, відповідно до обраних критеріїв	Виконано