МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

ІНСТИТУТ КОМП’ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА КОМП’ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Курсова робота

з дисципліни: «Мережній інформаційні системи»

на тему: «Система підтримки рішень для провайдера Інтернет послуг»

Виконав: Олійник О.С.

ІКІТ-512

Перевірила: Клімова А.С.

Київ 2014

Зміст

[Вступ 3](#_Toc405209169)

[Розробка програмної системи 4](#_Toc405209170)

[Архітектура системи 4](#_Toc405209171)

[Архітектура бази даних 5](#_Toc405209172)

[Функціональні та бізнес можливості системи 6](#_Toc405209173)

[Генерація звітів 10](#_Toc405209174)

[Висновок 12](#_Toc405209175)

[Список використаної літератури 13](#_Toc405209176)

# Вступ

Метою даної курсової роботи є проектування і створення системи, спрямованої на полегшення тяжкої і рутинної роботи викладачів, за рахунок централізованого зберігання даних про успішність і зручного доступу до них.

Розробка даної системи є досить важливою для покращення ефективності роботи університету. По-перше дана система спрощує взаємодію двох сторін – викладачів та деканатів. Спрощення відбувається в тому, що викладач може сам і з легкістю підключатися до БД і без труднощів додавати змінювати да. По-друге система організовую видачу інформацію так, що студент зможе сам ознайомитися з своїми результати в віддаленій формі.

# Розробка програмної системи

Для розробки даної системи використовувались наступні області:

* База даних (БД). Програмний засіб – Oracle XE.
* Бізнес рівень. Використовувалась мова програмування Java та сервер додатків Glassfish.
* Клієнтський рівень. Використовувались html, css, js, jquery, jsp, ajax.

## Архітектура системи

Архітектура системи (взаємодію клієнта, сервера, бази даних та різних API, які використовувались) представлена на рисунку 1.



Рис. 1. Архітектура системи.

Дана архітектура базується на шаблоні MVC (model view controller). Використання даного шаблону покращило читабельність коду, розбило систему на основні рівні – що зробило систему масштабованою та розширюваною. Також використання даного шаблону покращило безпеку.

# Робота з БД

Робота з БД відбувається за допомогою JDBC.

**Java DataBase Connectivity** ([англ.](http://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) Java DataBase Connectivity — *з'єднання з базами даних на*[*Java*](http://uk.wikipedia.org/wiki/Java)), скорочено **JDBC**) — [прикладний програмний інтерфейс](http://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%B8%D0%B9_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B8%D0%B9_%D1%96%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81) [Java](http://uk.wikipedia.org/wiki/Java), який визначає методи, з допомогою яких програмне забезпечення на Java здійснює доступ до [бази даних](http://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85). JDBC — це платформо-незалежний промисловий стандарт взаємодії Java-[застосунків](http://uk.wikipedia.org/wiki/%D0%97%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA) з різноманітними [СУБД](http://uk.wikipedia.org/wiki/%D0%A1%D0%A3%D0%91%D0%94), реалізований у вигляді пакета java.sql, що входить до складу [Java SE](http://uk.wikipedia.org/w/index.php?title=Java_SE&action=edit&redlink=1).

В основі JDBC лежить концепція так званих драйверів, що дозволяють отримувати з'єднання з базою даних по спеціально описанному [URL](http://uk.wikipedia.org/wiki/URL). Драйвери можуть завантажуватись динамічно (під час роботи програми). Завантажившись, драйвер сам реєструє себе й викликається автоматично, коли програма вимагає [URL](http://uk.wikipedia.org/wiki/URL), що містить протокол, за який драйвер «відповідає».

**Інтерфейси:**

JDBC [API](http://uk.wikipedia.org/wiki/API) містить два основні типи інтерфейсів: перший - для розробників [застосунків](http://uk.wikipedia.org/wiki/%D0%97%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA) і другий (нижчого рівня) - для розробників драйверів.

З'єднання з базою даних описується класом, що реалізує інтерфейс java.sql.Connection.

Маючи з'єднання з базою даних, можна створювати об'єкти типу Statement, використовувані для здійснення запитів до бази даних на мові [SQL](http://uk.wikipedia.org/wiki/SQL).

Існують такі види типів Statement, що відрізняються своїм призначенням:

* java.sql.Statement — Statement загального призначення;
* java.sql.PreparedStatement — Statement, що служить для здійснення запитів, котрі містять підставні параметри (позначаються символом '?' у тілі запиту);
* java.sql.CallableStatement — Statement, призначений для виклику збережених процедур.

Класjava.sql.ResultSet дозволяє легко обробляти результати запитів.

## Архітектура бази даних

База даних є одною із ключових частин будь-якої системи, тому правильність архітектури є дуже важливою для роботи системи. Нижче, на рисунку 2, представлена спрощена ER (entity relation) діаграма, яка показує поля таблиці та взаємодію між таблицями, які є в БД.

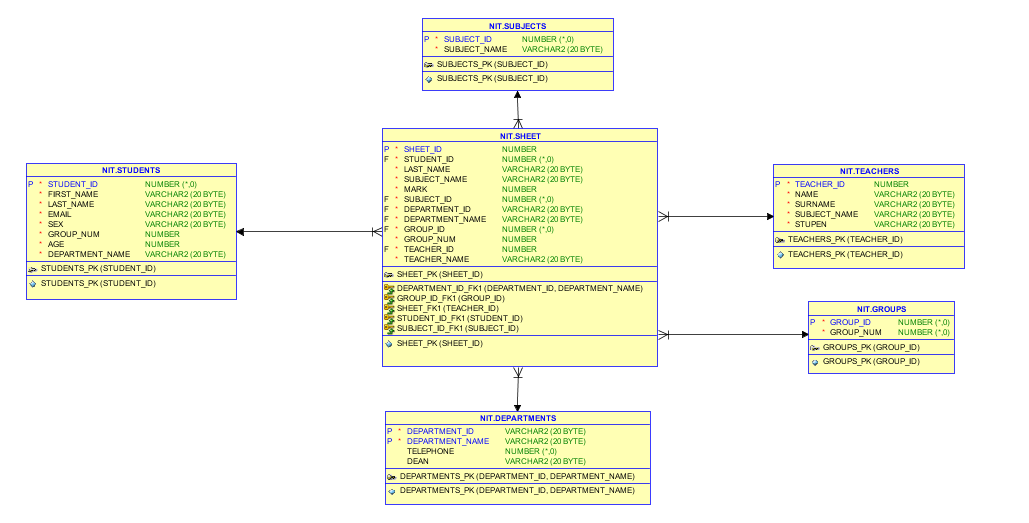


Рис. 2. ER-діаграма.

Далі приведено короткий опис таблиць БД:

|  |  |
| --- | --- |
| **Назва таблиці** | **Призначення** |
| NIT.SUBJECTS | Таблиця з назвами предметів |
| NIT.STUDENTS | Таблиця з персональними даними всіх студентів |
| NIT.TEACHERS | Таблиця з персональними даними всіх викладачі кафедр |
| NIT.DEPARTMENTS | Таблиця зі всіма даними про кожну з кафедр університету |
| NIT.GROUPS | Таблиця з поточними даними груп |
| NIT.SHEET | Таблиця з результатим студента і з логічними зв’язками між всіма таблицями |

# Функціональні можливості системи

В даній системі були реалызовані такі функції?

|  |  |  |
| --- | --- | --- |
| № | Назва функції | Призначення функції |
| 1 | Додавання студента | Занесення студента з відповідними даними до БД |
| 2 | Перегляд студентів | Перегляд інформації про студента |
| 3 | Видалення студента | Видалення даних студента відповідно заданого запросу |
| 3 | Занесення змін до студента | Заміна даних відповідно за запиту |
| 4 | Перегляд інформації з БД | Перегляд відповідних даних |
| 5 | Пошук по БД | Пошук інформації за певними критеріями |
| 6 | Редагування таблиць БД | Зміна дозволених даних відповідних таблиць БД |
| 7 | Видалення даних з БД | Видалення відповідних даних |

# Меню Сайту



Рис. 3. Меню сайту.

Home – головна сторінка сайту, де відображаються всі студенти.

Progress – відобраються дані про результати студента.

Tables має підменю : Students, Teachers, Shet. На даній сторінці відображається інформація відповідних таблиць з БД.

Change має підменю searchStudents, searchTeachers, searchTeachers. На даних сторінках дозволяється робота з певними таблицями БД.

AddStudent – Сторінка для додавання студенту.

Головний функціонал системи відображається в пункті меню Change.

Розглянемо кожне підменю Change:  
**1. searchStudents:**

Дана сторінка відображає роботу з таблициє студенти. Вона дозволяє виконувати наступні функції:

а) Add Student(рис.4) –Додати нового студента до БД \*



Рис. 4. Додати студента

b) Search Student(рис.5) – Пошук студента по будь-якому з його аргументів.

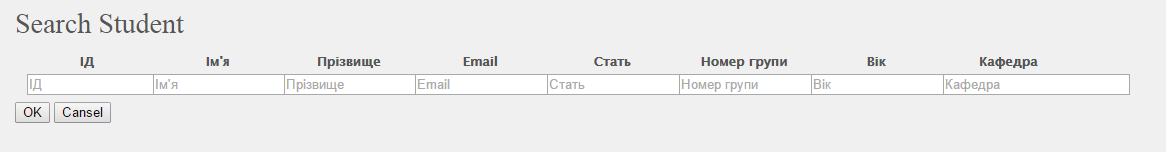


Рис. 5. Пошук студента

c) Update Student(рис.6) – оновлення даних про студента.



Рис. 6. Оновлення студента.

d) Delete Student (рис.7) – видалення студента.

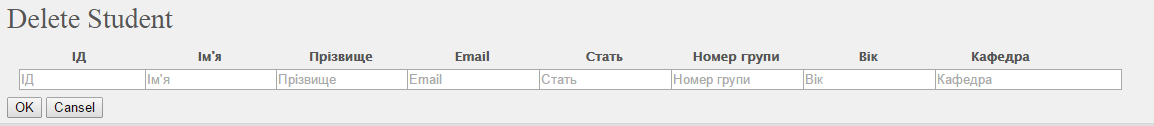


Рис. 7. Видалити студента

**2. searchTeachers:**

Дана сторінка відображає роботу з таблициє викладачі. Вона дозволяє виконувати наступні функції:

а) Add Teacher(рис.8) –Додати нового викладача до БД .ю.юю

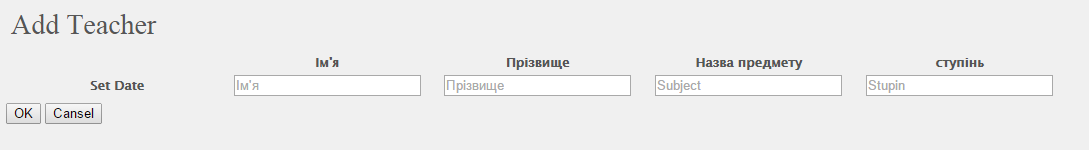


Рис. 8. Додати викладача

b) Search Teacher (рис.9) – Пошук викладача по будь-якому з його аргументів.

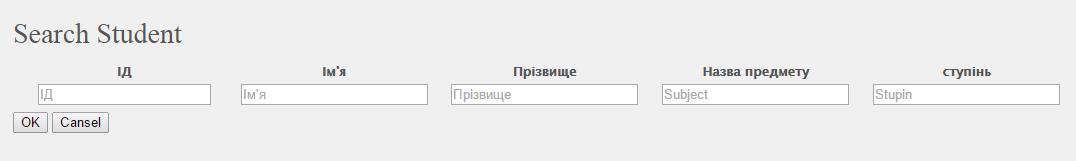


Рис. 9. Пошук викладача

c) Update Teacher(рис.10) – оновлення даних про викладача .

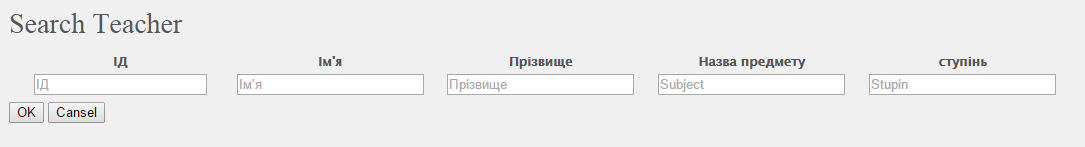


Рис. 10. Оновлення викладача.

d) Delete Teacher (рис.11) – видалення викладача .

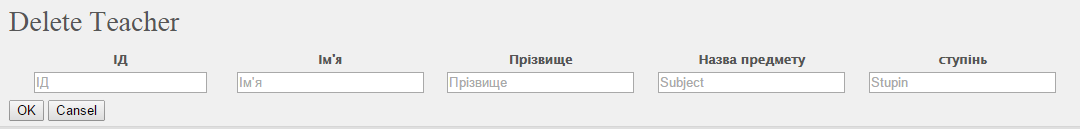


Рис. 11. Видалити викладача.

**3. searchSheet**

Сторінка дозволяє робити пошук по сторінці за відповідними параметрами.

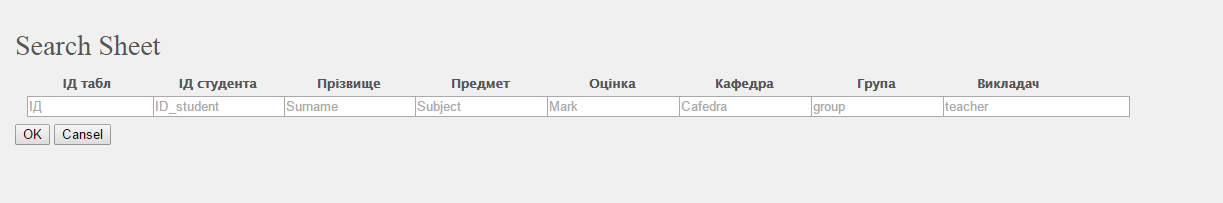


Рис.

Рис. 12. Пошук по Sheet.

**Додавання Студента**

Студента можна добавити двома способами:

1. В головному меню на сторінці addStudent (рис. 13) ;
2. В підменю Change в розділі searchStudent (рис. 14).

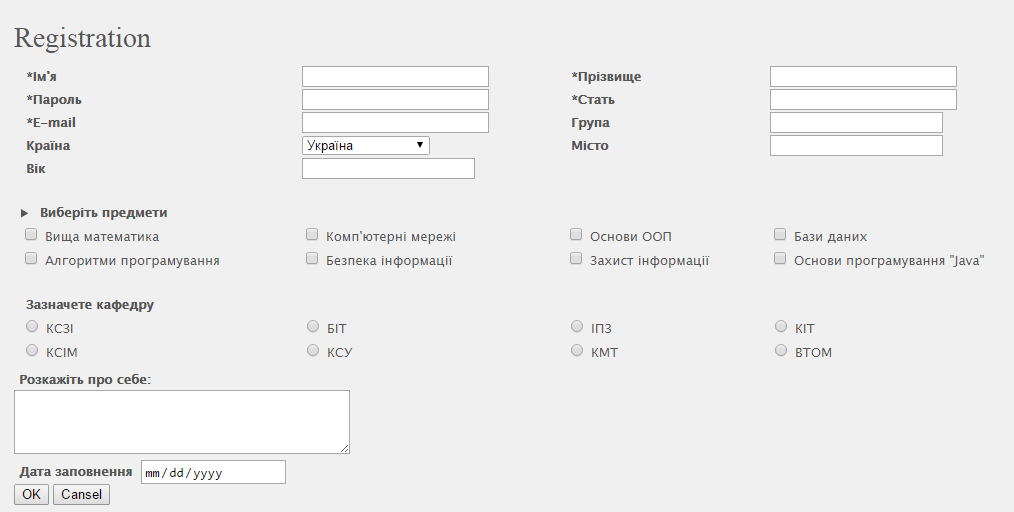


Рис. 13. Пошук по Sheet.

# Висновок

Основна мета курсового проекту реалізувати система, яка буде базуватись на веб-технологіях та використовувати бази даних для зберігання інформації. Дана ціль досягнута створенням системи обробки успішності студентів.

# Список використаної літератури

1. Документація java 1.7 <https://docs.oracle.com/javase/7/docs/>
2. Документація Oracle <https://docs.oracle.com/en/>
3. Документація Jersey <https://jersey.java.net/documentation/latest/user-guide.html>
4. Ralph Johnson, John Vlissides, Richard Helm, and Erich Gamma, “Design Patterns: Elements of Reusable Object-Oriented Software”, 1994
5. Bruce Eckel, “Thinking in Java”, 2006