УДК 004.75

АВТОМАТИЧНИЙ ОБЛІК ВІДВІДУВАНЯ ЗАНЯТЬ СТУДЕНТАМИ

Д. В. Чернов, К. В. Свідовська

У багатьох навчальних закладах ми бачимо плавний перехід на автоматизацію процесів навчання, за рахунок чого оптимізується навчальний процес та з'являється додатковий час на парі або уроці, як у студентів так і у викладачів. Даний web – додаток створений для автоматизації обліку відвідування занять студентами.

Мета роботи – розробити інформаційну систему, спроектувати базу даних в предметній області «Облік студентів», на основі якої розробити web-додаток для роботи з даними, основними функціями якого є: перегляд, редагування та створення пар, а також основних їх атрибутів (студент і предмет), робота зі звітами, відправлення їх електронною поштою.

Завдання дослідження – проаналізувати види та методи авторизації та аутентифікації, виявити основних користувачів та призначити їм класифікацію, на основі якої розробити інформаційну систему та під кожну з класифікацій розробити окремий сценарій роботи додатка. Для реалізації додатка використана логічна зв'язка MySQL + Java + MVC + DAO + Spring[1] + Hibernate[2] + FreeMarker.

Для реалізації рішення можна виділити основні етапи:

– Реалізація бази даних та її наповнення.
– Програмна реалізація додатка.
– Підключення додатка до розробленої бази даних.
– Налаштування багатокористувацького режиму.
– Налаштування авторизації та аутентифікації на всіх необхідних рівнях.
– Налаштування Logging – файлів.

Обраний метод аутентифікації та авторизації користувача в системі – пароль + підтвердження електронної адреси[1].

Метод аутентифікації на парі – qr-код після попередньої авторизації.

Алгоритм роботи програми:

– Аутентифікація та авторизація користувача в системі з розподіленням привілеїв.
– Користувач, якого визначено як «ADMIN» може створити пару, після чого автоматично генерується qr – запрошення для студентів (користувачів – «USER»).
– Користувач, якого визначено як «USER» сканує qr – запрошення та переходить на сторінку для підтвердження своєї присутності на парі.

Для кращого розуміння функціональності додатку з боку користувача наведено «use-case» діаграму на наступному рисунку (рис. 1).

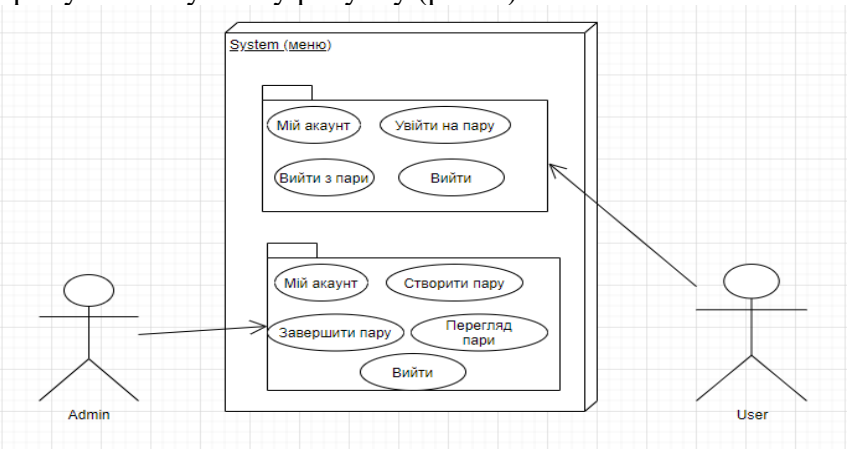

Рис. 1. Діаграма «use case» для панелі меню

В роботі спроектована база даних, яка відповідає умовам функціональності даного додатка. На рис. 2 показана ЕР-діаграма бази даних з основними сутностями і зв'язками між ними. Для створення схеми була проведена нормалізація до НФ3, результат якої можемо побачити на рис. 3. Ми маємо базу даних, яка складається з 10 таблиць. Так як в роботі використовується ORM Hibernate [2] та зв'язки між таблцями створені за допомогою анотацій, на діаграмі можна побачити лише таблиці.

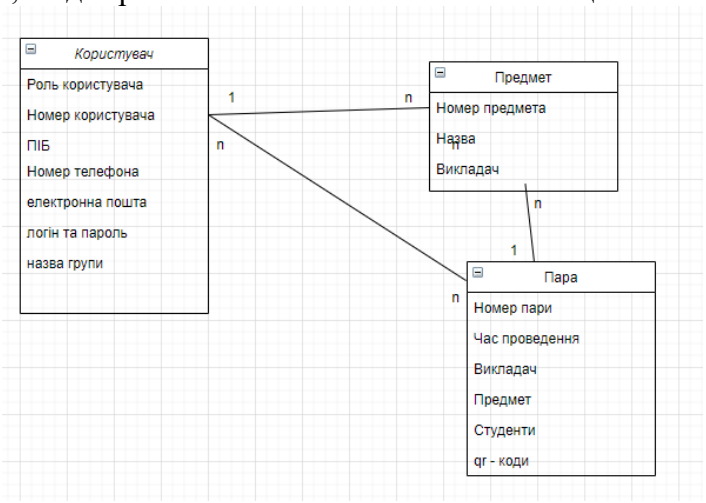

Рис. 2. ER-діаграма опису моделі даних

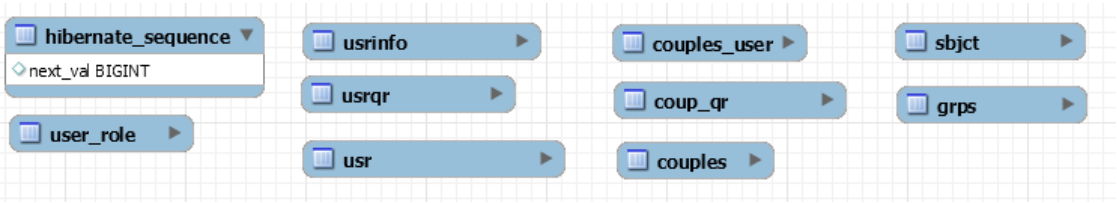

Рис. 3. Схема бази даних «classroom» після нормалізації до 3НФ

В роботі були розглянути методи аутентифікації та авторизації і в подальшому планується додати такі методи, як NFC, RFID, відбиток пальця або електронно цифровий підпис, так як варіант з qr-кодом не є повним вирішенням проблеми обліку студентів, а є першим етапом.

Недоліком обліку тільки за допомогою qr – коду є необхідність кожному студенту мати при собі пристрій читання (сканер) з підключенням до мережі інтернет. Також є вірогідність невірної ідентифікації за рахунок входу одного студента на декілька акаунтів (передача особистих даних від одного користувача додатка іншому) [3].

Основні панелі можна побачити на наступному рисунку (рис. 4).

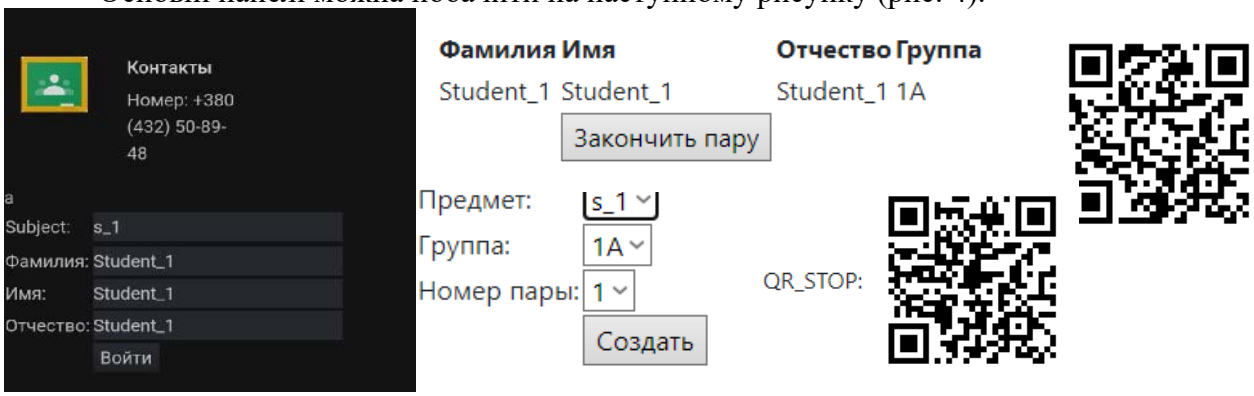

Рис. 4. Панелі керування

В роботі створений прототип інформаційної системи для обліку відвідування занять студентами, який дозволяє оптимізувати навчальний процес за рахунок автоматичного відвідування пар. В межах роботи створено web – додаток, основними функціями якого є: перегляд, редагування та створення пар, а також основних їх атрибутів (студент і предмет). Використано протокол IP.

Література

1. Документація Spring Framework. URL: https://docs.spring.io/spring-framework/docs/current/reference/html/
2. Документація ORM Hibernate. URL: https://hibernate.org/orm/documentation/5.4/
3. Методи аутентифікації та авторизації. URL: https://sites.google.com/site/identifikaciataautentifikacia/ponatta-pro-autentifikaciu/metodi-autentifikaciie

УДК 004.056: 004.492

ANTIVIRUS SOFTWARE VERSUS MALWARE

Harrison Asamoah

Antivirus software is a class of program designed to prevent, detect and remove malware infections on individual computing devices, networks and IT systems. It is originally designed to detect and remove viruses from computers, can also protect against a wide variety of threats, including other types of malicious software, such as

- Keyloggers
- Browser hijackers
- Trojan horses
- Worms
- Rootkits
- Spyware
- Adware
- Botnets
- Ransomware.

Antivirus software typically runs as a background process, scanning computers, servers or mobile devices to detect and restrict the spread of malware. Many antivirus software programs include real-time threat detection and protection to guard against potential vulnerabilities as they happen, as well as system scans that monitor device and system files looking for possible risks.

Antivirus software usually performs these basic functions:

- Scanning directories or specific files for known malicious patterns indicating the presence of malicious software.
- Allowing users to schedule scans so they run automatically.
- Allowing users to initiate new scans at any time; and
- Removing any malicious software, it detects. Some antivirus software programs do this automatically in the background, while others notify users of infections and ask them if they want to clean the files.

Malware, short for malicious software, is a blanket term for viruses, worms, trojans and other harmful computer programs hackers use to wreak destruction and gain access to sensitive information. It is the collective name for a number of malicious software variants, including viruses, ransomware and spyware. Malware is typically delivered in the form of a link or file over email and requires the user to click on the link or open the file to execute the malware. Malware will inevitably penetrate your network. You must have defenses that provide significant visibility and breach detection. In order to remove malware, you must be able to identify