

УДК (UDC) 004.94

**Стрілець
Вікторія Євгенівна**

к.т.н., доцент кафедри теоретичної та прикладної системотехніки,
Харківський національний університет імені В. Н. Каразіна, майдан
Свободи, 4, м. Харків, 61022
e-mail: viktoria.strilets@karazin.ua;
<https://orcid.org/0000-0002-2475-1496>

**Голіков
Максим Сергійович**

студент факультету комп'ютерних наук, Харківський національний
університет імені В. Н. Каразіна, майдан Свободи, 4, м. Харків,
61022
e-mail: xa12284048@student.karazin.ua;
<https://orcid.org/0000-0003-4842-7823>

Модель комп'ютерної системи поселення студентів у гуртожиток

Мета роботи: автоматизація процесу поселення студентів у гуртожитки університетів через створення комп'ютерної системи поселення, яка враховує особисті вимоги студентів та покращує розподіл місць.

Методи дослідження: методи оптимізації, зокрема еволюційний підхід з застосуванням генетичних алгоритмів, моделювання архітектури програмного забезпечення, методи та технології розробки програмного забезпечення. Комп'ютерна система побудована на основі трирівневої архітектури, яка включає рівні презентації, бізнес-логіки та даних. Для розробки використані технології: Spring Boot для серверної частини, Angular для клієнтської частини та PostgreSQL для зберігання даних.

У **результаті** розроблена комп'ютерна система дозволяє автоматизувати процес поселення студентів у гуртожитки, враховуючи їхні індивідуальні потреби та переваги. Вона забезпечує кращий розподіл студентів по кімнатах гуртожитків, з урахуванням таких критеріїв, як стать, тип особистості (інтроверт/екстраверт), курс навчання. Система забезпечує можливість адміністрування та моніторингу процесу поселення, що дозволяє працівникам університету оперативно реагувати на зміни та вирішувати можливі проблеми. Вибір технологій, таких як Spring Boot, Angular та PostgreSQL, забезпечує кросплатформність, безпеку даних та можливість адаптації системи до змінних умов. Модель бази даних включає таблиці, у яких зберігається інформація про студентів, працівників, кімнати гуртожитків та інші необхідні дані, що дозволяє забезпечити ефективне управління інформацією.

Висновки: запропонована система дозволяє автоматизувати процес розподілення студентів у кімнати гуртожитку, враховуючи їхні індивідуальні потреби та побажання. Це робить її перспективним інструментом для управління житловим фондом закладів вищої освіти, що може прискорити процес поселення та спростити його бюрократичну сторону. Автоматизація процесу поселення дозволяє зменшити навантаження на адміністративний персонал та мінімізувати кількість помилок, що виникають під час ручного розподілу місць у гуртожитках. Таким чином, розроблена система не тільки покращує умови оформлення документів на проживання для студентів, але й сприяє підвищенню ефективності роботи адміністративного персоналу університетів.

Ключові слова: комп'ютерна система, еволюційний підхід, генетичний алгоритм, оптимізаційна задача, поселення студентів.

Як цитувати: Стрілець В.Є., Голіков М.С. Модель комп'ютерної системи поселення студентів у гуртожиток. *Вісник Харківського національного університету імені В.Н.Каразіна, серія. «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»*. 2024. вип. 62. С.45-54. <https://doi.org/10.26565/2304-6201-2024-62-05>

How to quote: V.Y. Strilets, and M.S. Holikov, "Model of the computer system for settling students in a dormitory", *Bulletin of V.N. Karazin Kharkiv National University, series "Mathematical modelling. Information technology. Automated control systems*, vol. 62, pp.45-54, 2024. <https://doi.org/10.26565/2304-6201-2024-62-05>

1 Вступ

Стратегія розвитку сучасного суспільства спрямована на впровадження інформаційно-комунікаційних технологій у всі сфери життя. Цифровізація є державною політикою України [1], яка має на меті побудову і розвиток інформаційного суспільства, орієнтованого на інтереси людей та відкрите до доступу, обміну і користування даними. Цифровізація сприяє полегшенню доступу до послуг у сфері охорони здоров'я й освіти, наданню фінансових послуг, прозорості та ефективності процесів управління.

Створення цифрового простору в закладах вищої освіти дозволяє не тільки забезпечити ефективний навчальний процес, але й поліпшити взаємодію між студентами і відділами та службами університетів, які відповідають за життєзабезпечення. Однією з таких служб є служба поселення студентів, яка відповідає за розміщення іногородніх студентів, аспірантів, докторантів, та сімей студентів, аспірантів, докторантів у гуртожитках університетів. Щорічно комісії з поселення розглядають заяви студентів (аспірантів, докторантів) на надання місця проживання у гуртожитках. Підходи до розподілення студентів по кімнатах зазвичай враховують лише стать студентів і напрямок підготовки (академічні групи). Тоді як особистісні риси студентів і побажання залишаються без уваги. Тому задля вирішення цієї проблеми є можливим створення інформаційних систем для поселення студентів, які будуть приймати заявки від студентів і дозволять автоматизувати процес розселення з урахуванням різноманітних критеріїв, починаючи від кількості місць житлового фонду і закінчуючи побажаннями студентів щодо психотипу їхніх майбутніх сусідів.

Метою роботи є автоматизація процесу поселення студентів до гуртожитків через створення, реалізацію і впровадження моделі комп'ютерної системи поселення студентів.

Створювана комп'ютерна система передбачає наявність модуля, який буде відповідати за вирішення задачі розселення студентів. Дана задача є комбінаторною оптимізаційною задачею про призначення. У контексті поселення студентів необхідно до уваги різні вимоги, наприклад, проживання студентів однієї статі в кімнатах, процес взаємного впливу чи спілкування студентів з урахуванням їх типу особистості, кількість вільних місць у кімнатах тощо. Саме ці вимоги (критерії) формують множину обмежень задачі оптимізації. Для її розв'язання пропонується використати еволюційний підхід, зокрема генетичні алгоритми.

Еволюційні алгоритми, натхненні природним відбором і генетичними процесами, є дієвими в розв'язанні комбінаторних оптимізаційних задач [2, 3]. Однією із ключових причин вибору еволюційних алгоритмів, особливо генетичного алгоритму, є здатність адаптуватися до змінних умов. Наприклад, у задачі розселення студентів можуть виникати нові обмеження або вимоги, на які традиційні методи реагують повільно або недостатньо гнучко. Еволюційні алгоритми, завдяки своїй природі, здатні оперативнo корегувати свої рішення, забезпечуючи при цьому ефективність рішень.

Отже, впровадження еволюційного підходу в модель комп'ютерної системи поселення студентів обіцяє значні переваги: можливість врахування множини факторів різних типів, покращене використання ресурсів та гнучкість управління. Це робить створювану модель перспективним інструментом для вирішення організаційних і управлінських задач у сфері соціального забезпечення закладів вищої освіти.

2 Опис існуючих рішень і формулювання проблеми для розв'язання

Процес поселення студентів у гуртожиток регулюється нормативно-правовими документами навчальних закладів і є достатньо стандартним з незначними відмінностями, тобто автоматизовані системи поселення студентів і їх моделі будуть базуватися на схожих принципах. Таким чином, одними з основних вимог до подібних систем є кросплатформність і безпека даних [4].

Для побудови інтегрованого інформаційного середовища в закладах вищої освіти наразі створені і використовуються такі автоматизовані системи, як АС Деканат, АС Приймальна комісія, АС Студмістечко [5]. Дані системи розроблені Науково-дослідним інститутом прикладних технологій (м. Київ) і успішно впроваджуються у роботу ВНЗ. Вони відповідають вимогам до кросплатформності і безпеки даних, але їх вартість є достатньо високою, а використання потребує попередніх тренінгів для адміністративного персоналу, щоб набутися необхідних навичок для роботи з системами.

Також наразі існує модель автоматизації поселення студентів для Національного університету «Львівська політехніка», розроблена за допомогою HiAsm, інтегрованого середовища розробки програм для Windows [6]. Ця модель забезпечує повний автоматизований процес поселення студентів у гуртожиток. Однак, оскільки вона розроблена тільки для однієї платформи, не можна вважати її повністю кросплатформною.

Крім того, існуючі автоматизовані системи реалізують типовий підхід до розміщення студентів, без урахування додаткових особистих вимог. Тому пропонується розробити модель комп'ютерної системи поселення студентів, яка відповідає вимогам кросплатформності та

безпеки, а також здатна отримати оптимальне розміщення студентів, приймаючи до уваги низку додаткових факторів (наприклад, психотип людини або релігійні переконання).

3 Еволюційний підхід для розв'язання задачі про розподілення студентів в гуртожитку

У системах автоматизації поселення студентів у гуртожиток потрібно врахувати різні критерії і вимоги. Більшість критеріїв можуть бути суперечливими, тому виникає проблема вибору коректного критерію. Це означає, що доводиться йти на компроміс, обираючи для кожного критерію значення, яке не є оптимальним, але прийнятним з урахуванням інших критеріїв.

У створеній моделі поселення студентів пропонується взяти до уваги такі критерії: стать, тип особистості (інтроверт чи екстраверт) та курс студентів.

1. Гендерний критерій:

- для кожної кімнати перевіряється, чи всі студенти однієї статі (чоловіки або жінки);
- якщо в кімнаті є і чоловіки, і жінки, збільшується лічильник помилки невідповідності статі *genderMismatch*, який показує кількість таких кімнат.

2. Баланс інтровертів та екстравертів:

- обчислюється кількість інтровертів та екстравертів у кожній кімнаті;
- якщо кількість інтровертів і екстравертів однакова, показник невідповідності балансу *sociotypeMismatch* буде збільшуватися;
- перевіряється, чи в половині кімнат цей баланс рівний, і якщо так, то *sociotypeMismatch* збільшується ще раз.

3. Розподіл за роками навчання:

- підраховується кількість студентів кожного року навчання в кімнаті та обчислюється середня різниця між роками;
- якщо різниця між роками для будь-якого студента перевищує середню різницю більше, ніж на один курс, збільшується помилка середньої різниці року навчання *courseMismatch*.

Загальна цільова функція обчислюється за формулою:

$$f = 0.8\text{genderMismatch} + 0.1\text{sociotypeMismatch} + 0.1\text{courseMismatch}.$$

Для даної цільової функції потрібно досягти мінімуму, оскільки в такому випадку буде забезпечена мінімальна помилка для кожного з критеріїв. Чим менше значення цільової функції, тим ефективнішим є розподілення студентів.

Вагові коефіцієнти для кожного критерію визначають їх важливість в загальній оцінці цільової функції. Найбільший вплив має гендерний критерій, оскільки це важливо для комфорту проживання студентів.

Для розв'язання сформульованої задачі про розподілення студентів в гуртожитку був обраний еволюційний підхід, а саме – генетичний алгоритм.

Еволюційний підхід використовує принципи еволюційних процесів у природі для вирішення проблем пошуку найкращих рішень серед можливих варіантів. Цей підхід базується на теорії природного відбору Дарвіна та концепції генетичного розвитку. Еволюційні алгоритми – це потужний інструмент для вирішення будь-яких складних задач. Цей тип алгоритмів включає в себе генетичні алгоритми, еволюційну стратегію, еволюційне програмування, алгоритми диференціальної еволюції та генетичне програмування. Їх широко використовують у різних галузях, включаючи оптимізацію, штучний інтелект та інженерію [7].

Генетичні алгоритми використовують концепції "хромосом" і "генів" для моделювання можливих розв'язків задачі. Хромосоми представляють розв'язки, а гени визначають їх параметри чи характеристики. Наприклад, для задачі поселення студентів у кімнати, кожна хромосома може представляти собою список кімнат, а гени визначатимуть кожну кімнату у цьому списку, і кожен студент у кімнаті – це значення гену [8].

Важливо зазначити, що генетичні алгоритми працюють з популяцією індивідів, що відрізняється від інших алгоритмів, які працюють з одним рішенням.

Генетичний алгоритм імітує природний відбір, створюючи та еволюціонуючи популяцію рішень, де кращі рішення мають більше шансів на розмноження та мутацію. В розглядуваній задачі рішення – це розподіл студентів по кімнатах, враховуючи їхні характеристики, такі як стать, курс і соціотип.

Основна ідея полягає в тому, що випадкові мутації та схрещування створюють нові покоління рішень, які все краще вирішують поставлену задачу. В алгоритмі реалізовано адаптацію

параметрів (ймовірності мутації та кросовера) на основі середньої пристосованості популяції, що дозволяє алгоритму швидше знаходити наближені до оптимальних рішення, адаптуючись до змінних умов.

Алгоритм складається з п'яти основних етапів у заданому порядку:

- ініціалізація;
- відбір;
- схрещування;
- мутація;
- оцінка пристосованості.

На етапі ініціалізації потрібно визначити вхідні дані, щоб створити початкову популяцію. Це здійснюється шляхом запитів до бази даних для отримання списку студентів і кімнат з їх кількістю місць. Дані про студентів включають стаття, курс та соціотип, а також інформацію про кількість кімнат і кількість місць у кожній кімнаті.

Після отримання даних, студенти випадковим чином розподіляються по кімнатах, створюючи початкову популяцію. Генотип, у цьому контексті, представляє структуру даних, яка визначає розподіл студентів по кімнатах. Встановлюється розмір популяції.

Обчислення пристосованості (цільової функції) кожного індивіда відбувається після створення популяції. Вони сортуються за рівнем пристосованості, і перша половина популяції відбирається як потенційні батьки. Метод відбору, який використовується, називається "турнірний відбір". Його суть полягає в тому, що краще пристосовані особини мають більше шансів передати свої "гени" наступному поколінню, покращуючи таким чином якість рішень.

Після відбору відбувається схрещування. Використовується одноточковий кросовер, де випадковим чином вибирається точка розрізу в генетичних послідовностях (списку кімнат). Частина від одного батька передається першому нащадку до точки розрізу, а частина від іншого батька після точки розрізу – другому нащадку.

Потім відбувається мутація, контрольована ймовірнісним методом. Використовується "мутація обміну", де двоє студентів змінюються місцями між різними кімнатами, щоб зберегти корисні комбінації.

Адаптивність кросовера та мутації – особлива риса цього алгоритму. Вона дозволяє алгоритму адаптуватися до умов задачі та підтримувати баланс між дослідженням нових рішень (exploration) і використанням вже знайдених (exploitation). Якщо кросовер добре працює, його ймовірність збільшується, і навпаки, якщо мутація ефективніша, то збільшується її ймовірність.

Такий підхід до розв'язання задачі дозволяє алгоритму добре пристосовуватися і ефективно знаходити наближені до оптимального рішення в змінному середовищі.

4 Моделювання процесу поселення студентів

При створенні моделі комп'ютерної системи необхідно визначити загальну структуру та напрям функціонування системи, які мають відображати основні етапи процесу поселення студентів у гуртожиток

Були виділені такі основні етапи цього процесу:

1. Реєстрація студента у системі.
2. Авторизація всіх користувачів, задіяних у процесі.
3. Студент надсилає запити працівникам університету для підтвердження виконання певних вимог, залежно від спеціалізації працівника.
4. Співробітники приймають рішення щодо підтвердження виконання конкретних вимог студентом.
5. Якщо студент виконав усі вимоги, він отримує дозвіл на поселення в гуртожиток.
6. Розподілу студентів по гуртожитках (кімнатах) з використанням генетичного алгоритму.
7. Додаткова перевірка на гендер, якщо це необхідно.

Детальніше розглянемо етапи 3 і 4. Під працівниками розуміються користувачі з наступними ролями:

- медсестра;
- економіст;
- заступник декана.

Кожен із цих співробітників відповідає за конкретні завдання: медсестра – підтверджує проходження студентом медичного огляду, економіст – перевіряє оплату місця в гуртожитку, а заступник декана – підтверджує статус студента. Цей процес можна описати на концептуальному рівні за допомогою діаграми прецедентів, як показано на рис. 1.

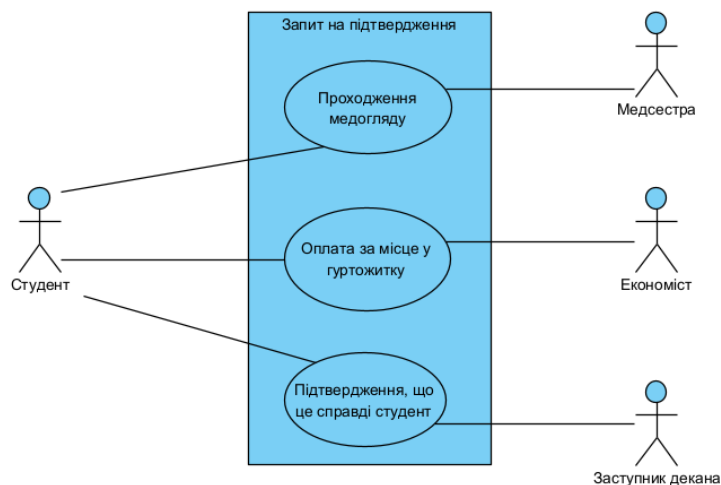


Рисунок 1. Діаграма прецедентів взаємодії студента й адміністрації

Таким чином студент через систему подає запит на поселення і необхідні для цього документи і отримує відповідь після розгляду документів адміністрацією і застосування моделі поселення.

5 Проектування архітектури комп'ютерної системи поселення студентів

Для забезпечення стабільного функціонування і безпеки даних для комп'ютерної системи поселення студентів була обрана розподілена архітектура.

Розподілена архітектура полягає в тому, що компоненти системи розташовані на різних фізичних або логічних машинах і з'єднані мережею для підвищення масштабованості та ефективності. Трирівнева архітектура є підтипом розподіленої архітектури, де програмна система розділена на три рівні або компоненти [9]. Кожен з них має своє призначення та спрямований на вирішення певних завдань. Ці рівні включають презентаційний, бізнес-логіки та рівень даних (рис. 2).

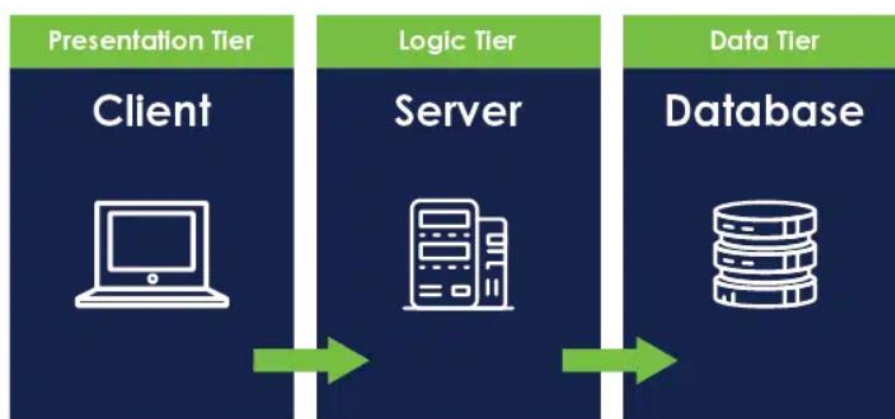


Рисунок 2. Модель трирівневої архітектури комп'ютерної системи

На рівні представлення знаходиться інтерфейс, який користувач бачить і з яким він взаємодіє. Цей рівень відповідає за виведення інформації для користувача та обробку його введень. Він передає команди для обробки наступним рівнем, а також отримує відповіді для відображення користувачеві.

Бізнес-логіка – це місце, де знаходиться "мозок" програмної системи. Тут обробляється бізнес-логіка та процеси, що визначають, як система має працювати. Цей рівень встановлює

контрольований та безпечний спосіб взаємодії між рівнем представлення та рівнем даних. Коли користувач взаємодіє з системою, наприклад, відправляє заповнену форму, бізнес-логіка аналізує цю інформацію та виконує потрібні операції для взаємодії з даними.

Рівень доступу до даних – це рівень, який передбачає обмін інформацією з базою даних чи іншими джерелами даних. Цей рівень відповідає за управління та доступ до даних для їх наступної обробки на шарі бізнес-логіки.

На рівні бізнес-логіки реалізований модуль для розв'язання задачі поселення студентів на основі запропонованого еволюційного підходу. Цей блок взаємодіє з базою даних та отримує вхідні дані про студентів: курс, соціотип та гендер. Після роботи алгоритму результати виводяться у консоль у спрощеному форматі, який показує кінцевий варіант поселення студентів по кімнатах. Цей підхід може бути розширений у майбутньому для створення повноцінної системи, яка буде враховувати більше параметрів та надавати користувачам більше інформації про їхнє майбутнє проживання.

6 Визначення структури бази даних комп'ютерної системи

Для формування структури бази даних перш за все, потрібно визначити головні сутності, з якими відбуватимуться основні операції. Такими сутностями є:

- student (таблиця з інформацією про студентів);
- users (таблиця для збереження даних користувачів);
- role (таблиця для визначення ролі користувача);
- employee (таблиця з інформацією про працівників університету);
- administrator (таблиця з даними про адміністраторів);
- dormitory (таблиця з інформацією про гуртожитки).

Додаткові таблиці використовуються для допоміжних цілей, щоб краще впорядкувати та структурувати дані. Наприклад, таблиці group, speciality і department доповнюють інформацію про приналежність студента до академічної групи і факультету, а таблиця room містить дані про кімнати гуртожитків. Загалом знадобилося 13 таблиць.

На основі концептуальної моделі бази даних побудована діаграма розширеної сутнісно-зв'язкової моделі (Enhanced Entity-Relationship, EER). Це розширення типової моделі сутнісно-зв'язкових діаграм додає можливості та додаткові концепції для повного і точнішого опису між об'єктами в базі даних саме взаємозв'язків [10]. Фрагмент цих зв'язків представлений у канонічній нотації EER-моделі на рис. 4.

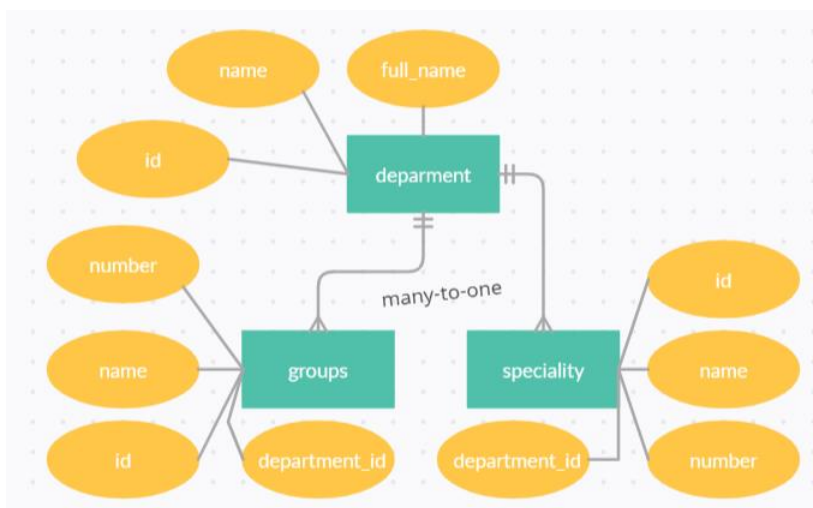


Рисунок 3. Фрагмент канонічної нотації EER-моделі для таблиць department, groups та speciality

7 Вибір компонентів для розробки

Щоб створене програмне забезпечення працювало на різних пристроях і платформах, необхідно використовувати технології, що забезпечують кросплатформність на рівні середовищ виконання. Одним із прикладів таких технологій є програмування на мові Java [11]. Ідеальним

рішенням була б модель, яка працює безпосередньо в браузері, без необхідності завантаження на пристрій [4].

Програмне рішення у форматі веб-додатку можна реалізувати за допомогою Spring Boot, Angular та PostgreSQL. Це дозволить використовувати комп'ютерну систему на різних пристроях, задовольняючи головний критерій — кросплатформність. У додатку також можна забезпечити безпечне зберігання даних студентів у хмарному сховищі, що є надійнішим, ніж тримання їх на окремому комп'ютері [4].

Під час розробки моделі комп'ютерної системи для вирішення задачі поселення студентів у гуртожиток важливо правильно обрати компоненти та інструменти, щоб забезпечити ефективність, надійність та безпеку додатку. Розглянемо переваги обраних компонентів:

1. Фреймворк Spring Boot ідеально підходить для розробки додатків завдяки своїй простоті та швидкому налаштуванню. Він має вбудований веб-сервер Tomcat, що полегшує розгортання додатків, а також пропонує багато корисних бібліотек і функцій, які заощаджують час і зусилля розробників [12, 13]. Його концепція "конвенції над конфігурацією" вказує, що багато параметрів вже вбудовані за замовчуванням, що дозволяє зосередитися на розробці коду, який вирішує конкретні завдання. Вбудований контейнер для обробки HTTP-запитів дозволяє запускати додатки як окремі виконувані файли, що полегшує процес розгортання

2. PostgreSQL – це найкращий вибір для бази даних з кількох причин:

- надійність: PostgreSQL славиться своєю стабільністю і надійністю, що забезпечує безпеку даних студентів та гуртожитків;
- розширюваність та гнучкість: можливість додавання власних типів даних, функцій та розширень дозволяє створювати унікальні рішення;
- підтримує обробку географічної інформації та роботи з геоданими завдяки розширенням, таким як PostGIS;
- відкритий код PostgreSQL сприяє створенню спільноти розробників і забезпечує повний контроль над базою даних;
- сумісність з Java та Hibernate: PostgreSQL добре працює з Java та ORM-рішеннями, такими як Hibernate, що полегшує взаємодію з базою даних із веб-додатку [14, 15].

3. ORM (Java Persistence API з Hibernate) дозволяє представляти об'єкти Java у вигляді даних у базі даних. Hibernate спрощує взаємодію з базами даних, дозволяючи виконувати операції без написання складних SQL-запитів [13].

4. Spring MVC – цей компонент Spring Framework використовує архітектурний шаблон Model-View-Controller, що полегшує розробку та управління веб-запитами у додатках на Java [12, 13].

5. Spring Security забезпечує безпеку та автентифікацію користувачів, захищаючи особисту та конфіденційну інформацію [12, 13].

6. Maven – популярний інструмент для складання проектів та управління залежностями у середовищі Java, що спрощує додавання бібліотек, забезпечує уніфікований процес зборки, дозволяє швидко впроваджувати нові функції та оновлення [16].

7. Angular – потужний фреймворк для розробки користувацьких інтерфейсів, що дозволяє створювати динамічні та інтерактивні веб-сайти [17]. Однією з головних переваг Angular є двостороння прив'язка даних, що автоматично відображає зміни в моделі в інтерфейсі користувача і навпаки. Angular також має потужний механізм для обробки HTTP-запитів та інтегровану систему модулів, що спрощує організацію коду і додавання нових функцій.

Таким чином, вибір цих компонентів обумовлений їх продуктивністю, ефективністю, масштабованістю та простотою використання.

8 Застосування моделі комп'ютерної системи поселення студентів

Створена програмна реалізація моделі комп'ютерної системи поселення студентів була протестована для розселення 112 студентів по 28 кімнатах гуртожитку. Кожна кімната мала 4 вільних місця.

При ініціалізації ці початкові дані були надані генетичному алгоритму. Розмір популяції заданий – 100 індивідів.

Для адаптації алгоритму до умов задачі були задані параметри операторів кросовера та мутації (табл. 1).

Таблиця 1. Використані параметри

Значення	Мутація	Кросовер
Мінімальне	0.01	0.6
Максимальне	0.5	0.9
Початкове	0.1	0.8

Алгоритм виконувався протягом 10000 ітерацій.

У результаті роботи алгоритму був отриманий розподіл студентів по кімнатах (рис. 4, 5).

```

Кімната 1:
Гендер: чоловік, Курс: 6, Соціотип: інтроверт
Гендер: чоловік, Курс: 5, Соціотип: інтроверт
Гендер: чоловік, Курс: 6, Соціотип: інтроверт
Гендер: чоловік, Курс: 5, Соціотип: екстраверт

Кімната 2:
Гендер: чоловік, Курс: 3, Соціотип: інтроверт
Гендер: чоловік, Курс: 3, Соціотип: інтроверт
Гендер: чоловік, Курс: 5, Соціотип: інтроверт
Гендер: чоловік, Курс: 3, Соціотип: екстраверт

Кімната 3:
Гендер: чоловік, Курс: 5, Соціотип: інтроверт
Гендер: чоловік, Курс: 3, Соціотип: інтроверт
Гендер: чоловік, Курс: 6, Соціотип: інтроверт
Гендер: чоловік, Курс: 1, Соціотип: інтроверт

Кімната 4:
Гендер: жінка, Курс: 3, Соціотип: інтроверт
Гендер: жінка, Курс: 4, Соціотип: екстраверт
Гендер: жінка, Курс: 6, Соціотип: інтроверт
Гендер: жінка, Курс: 1, Соціотип: інтроверт

Кімната 5:
Гендер: чоловік, Курс: 6, Соціотип: інтроверт
Гендер: чоловік, Курс: 3, Соціотип: інтроверт
Гендер: чоловік, Курс: 6, Соціотип: інтроверт
Гендер: чоловік, Курс: 1, Соціотип: інтроверт

```

Рисунок 4. Фрагмент результатів поселення

```

Кімната 16:
Гендер: чоловік, Курс: 2, Соціотип: інтроверт
Гендер: жінка, Курс: 1, Соціотип: екстраверт
Гендер: чоловік, Курс: 5, Соціотип: екстраверт
Гендер: жінка, Курс: 5, Соціотип: екстраверт

Кімната 17:
Гендер: жінка, Курс: 2, Соціотип: інтроверт
Гендер: чоловік, Курс: 4, Соціотип: екстраверт
Гендер: жінка, Курс: 1, Соціотип: інтроверт
Гендер: жінка, Курс: 1, Соціотип: екстраверт

Кімната 18:
Гендер: жінка, Курс: 4, Соціотип: інтроверт
Гендер: жінка, Курс: 5, Соціотип: інтроверт
Гендер: чоловік, Курс: 4, Соціотип: екстраверт
Гендер: чоловік, Курс: 1, Соціотип: інтроверт

```

Рисунок 5. Студенти з різними гендерами у кімнаті

Аналіз результатів розселення показує, що алгоритм працює і задача вирішується, але іноді може виникати ситуація, коли в одній кімнаті опиняються студенти різної статі (рис. 5). Для вирішення цієї проблеми буде розроблений додатковий алгоритм, який шукає студентів, що підходять для переміщення, і змінює їх місцями, забезпечуючи правильний розподіл.

9 Висновки

Процес цифровізації вимагає створення різноманітних інформаційних систем, до яких можна віднести запропоновану модель комп'ютерної системи поселення студентів.

Впровадження в комп'ютерну систему генетичного алгоритму, дозволяє удосконалити процес поселення студентів у гуртожитки, забезпечуючи розв'язання задачі розподілення студентів як задачі комбінаторної оптимізації. Використання еволюційного підходу забезпечує ефективний розподіл студентів за рахунок адаптивності до змінних умов і здатності швидко коригувати рішення у відповідь на нові обмеження чи вимоги. Це забезпечує виконання обмежень і вимог задачі, ефективне використання ресурсів та більшу гнучкість управління процесом поселення.

Для успішної реалізації моделі комп'ютерної системи був проведений детальний аналіз доступних підходів і технологій розробки. Були обрані для програмної реалізації технології, такі як Java, Spring Boot, Angular і PostgreSQL, що забезпечує кросплатформність, надійність і безпеку розроблюваного програмного забезпечення.

Розроблена архітектура комп'ютерної моделі з тривірневою структурою, гарантує ефективний розподіл обов'язків та кращу взаємодію між компонентами моделі. Впровадження еволюційних

алгоритмів на рівні бізнес-логіки дозволяє автоматизувати процес поселення з урахуванням гендерних, соціотипних та інших індивідуальних характеристик студентів.

Запропонований підхід має потенціал значно спростити та оптимізувати процес поселення студентів у гуртожитки, підвищити задоволеність користувачів і ефективність використання житлових ресурсів навчальних закладів.

СПИСОК ЛІТЕРАТУРИ

1. Концепція розвитку цифрової економіки та суспільства України на 2018 – 2020 роки / Розпорядження Кабінету Міністрів України від 17 січня 2018 р.
2. Wilson J. A genetic algorithm for the generalised assignment problem. *J Oper Res Soc* 48, 1997. pp. 804–809. DOI: <https://doi.org/10.1057/palgrave.jors.2600431>
3. Васильєв О. Б., Васильєва Н. С., Кічмаренко О. Д. Методи розв'язування задач багатокритеріальної оптимізації : метод. вказ. Одеса: Одеський національний університет імені І. І. Мечникова, 2017. 48 с.
4. Голіков М.С. Аналіз комп'ютерних моделей автоматизації поселення студентів у гуртожиток. *Комп'ютерне моделювання у наукоємних технологіях : зб. наукових праць VIII Міжнар. наук.-техн. конф.* Харків : ХНУ імені В.Н. Каразіна, 2022. С. 32- 34.
5. АСУ «ВНЗ». Головна сторінка : веб-сайт. URL: <https://vuz.osvita.net/>.
6. Струк Є., Дубленич Р., Фабрі Л., Автоматизація процесу поселення студентів у гуртожиток студмістечка. *Вісник національного університету «Львівська політехніка» Комп'ютерні науки та інформаційні технології.* 2016. Серія №843. С. 35-42.
7. Корнієнко В.І., Гусєв О.Ю., Герасіна О.В. Інтелектуальне моделювання нелінійних динамічних процесів у системах керування, кібербезпеки, телекомунікацій : підручник. Нац. техн. ун-т «Дніпровська політехніка». Дніпро : НТУ «ДП», 2020. 536 с.
8. Кононюк А.Ю. Нейронні мережі і генетичні алгоритми. Київ : Корнійчук, 2008. 446 с.
9. Three-Tier Architecture Approach for Custom Applications. IT Goals Achieved with Zircous, Technology Solutions : веб-сайт. URL: <https://www.zircous.com/2022/11/15/three-tier-architecture-approach-for-custom-applications-2/> (дата звернення: 12.04.2024).
10. Visual-paradigm. Diagrams : веб-сайт. URL : <https://www.visual-paradigm.com/> (дата звернення: 15.04.2024).
11. Bruce Eckel. Thinking in Java. 4th ed. p. cm. Includes bibliographical references and index : книга. United States of America : Courier in Stoughton, 2006. 1057 p.
12. Spring. Spring makes Java simple : веб-сайт. URL : <https://spring.io/> (дата звернення: 01.04.2024).
13. Baeldung. Learn Java : веб-сайт. URL : <https://www.baeldung.com/> (дата звернення: 01.04.2024).
14. PostgreSQL. База даних : веб-сайт. URL : <https://www.postgresql.org/> (дата звернення: 01.04.2024).
15. Пасічник В.В., Резніченко В.А. Організація баз даних та знань : підручник МОНУ. Київ: BHV, 2006. 384 с.
16. Apache Maven. Software project management and comprehension tool : веб-сайт. URL : <https://maven.apache.org/> (дата звернення: 01.04.2024).
17. Angular. Deliver web apps with confidence : веб-сайт. URL : <https://angular.io/> (дата звернення: 01.04.2024).

REFERENCES

1. Concept of the development of the digital economy and society of Ukraine for 2018-2020 / Decree of the Cabinet of Ministers of Ukraine dated January 17, 2018. [in Ukrainian]
2. J. Wilson. A genetic algorithm for the generalised assignment problem. *J Oper Res Soc* 48, 1997. pp. 804–809. <https://doi.org/10.1057/palgrave.jors.2600431>
3. O.B. Vasiliev, N.S. Vasilieva, O.D. Kichmarenko. *Methods for solving multicriteria optimisation problems: methodological guidance.* Odesa: Odesa National University named after I. I. Mechnikov, 2017. 48 p. [in Ukrainian]
4. M.S. Holikov. Analysis of computer models for automation of student accommodation in a hostel. *Computer modelling in science-intensive technologies: proceedings of the VIII International Scientific and Technical Conference: V. N. Karazin Kharkiv National University*, 2022. P. 32-34. [in Ukrainian]

5. ASU «VNZ». Main page: website. URL: <https://vuz.osvita.net/>.
6. E. Struk, R. Dublenych, L. Fabri. Automation of the process of settling students in the campus dormitory. *Bulletin of the National University 'Lviv Polytechnic' Computer Science and Information Technology*, Series No. 843, pp. 35-42, 2016. [in Ukrainian]
7. V.I. Korniienko, O.Y. Gusev, O.V. Gerasina. *Intellectual modelling of nonlinear dynamic processes in control systems, cybersecurity, telecommunications: textbook*. National Technical University 'Dnipro Polytechnic'. Dnipro: NTU 'DP', 2020. 536 c. [in Ukrainian]
8. A.Y. Kononiuk. *Neural networks and genetic algorithms*. Kyiv: Korniychuk, 2008. 446 c. [in Ukrainian]
9. Three-Tier Architecture Approach for Custom Applications. IT Goals Achieved with Zirous, Technology Solutions: website. URL: <https://www.zirous.com/2022/11/15/three-tier-architecture-approach-for-custom-applications-2/> (accessed 12.04.2024).
10. Visual-paradigm. Diagrams: website. URL: <https://www.visual-paradigm.com/> (accessed 15.04.2024).
11. Bruce Eckel. Thinking in Java. 4th ed. p. cm. Includes bibliographical references and index : book. United States of America: Courier in Stoughton, 2006. 1057 c.
12. Spring. Spring makes Java simple: website. URL: <https://spring.io/> (accessed 01.04.2024).
13. Baeldung. Learn Java : website. URL : <https://www.baeldung.com/> (accessed 01.04.2024).
14. Postgresql. Database : website. URL: <https://www.postgresql.org/> (accessed 01.04.2024).
15. V.V. Pasichnyk, V.A. Reznichenko. *Organisation of databases and knowledge: textbook of the Ministry of Education and Science*. Kyiv: BHV, 2006. 384 c. [in Ukrainian]
16. Apache Maven. Software project management and comprehension tool : website. URL: <https://maven.apache.org/> (accessed 01.04.2024).
17. Angular. Deliver web apps with confidence : website. URL: <https://angular.io/> (accessed 01.04.2024).

Strilets Viktoriia *Ph.D, associate professor of the theoretical and applied system engineering department, V.N. Karazin Kharkiv National University, 4 Svobody sq., Kharkiv, Ukraine, 61022*

Holikov Maksym *student of the computer science faculty V.N. Karazin Kharkiv National University, 4 Svobody sq., Kharkiv, Ukraine, 61022*

Model of the computer system for settling students in a dormitory

Purpose: to automate the process of settling students in university dormitories by creating a computer system that takes into account the personal requirements of students and improves the allocation of places.

Research methods: optimization methods, in particular evolutionary approach using genetic algorithms, software architecture modeling, software development methods and technologies. The computer system is built on a three-tier architecture that includes presentation, business logic, and data layers. Technologies used for development: Spring Boot for the server part, Angular for the client part and PostgreSQL for data storage.

As a **result**, the developed computer system allows automating the process of settling students in dormitories, considering their individual needs and preferences. It ensures better allocation of students to dormitory rooms, taking into account such criteria as gender, personality type (introvert/extrovert), study year and other important factors. The system also provides the ability to administer and monitor the settlement process, which allows university staff to quickly respond and solve possible problems. The choice of technologies, such as Spring Boot, Angular and PostgreSQL, ensures cross-platform compatibility, data security and the ability to adapt the system to changing conditions. The database model includes tables that store information about students, staff, dorm rooms and other necessary data, which allows for efficient information management.

Conclusions: the proposed system allows automating the process of assigning students to dormitory rooms, taking into account their individual needs and wishes. This makes it a promising tool for managing the residential fund of higher education institutions, which can speed up the settlement and simplify bureaucratic processes. Automation of the settlement process allows reducing the burden on administrative staff and minimize the number of errors that occur during the manual allocation of places in dormitories. Thus, the developed system not only improves the conditions for registration of residence documents for students, but also contributes to increasing the efficiency of the administrative staff of universities.

Keywords: computer system, evolutionary approach, genetic algorithm, optimization problem, student accommodation.