Марченко Олена Іванівна,

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ

ORCID ID 0000-0001-5754-4920

Чуй Олег Володимирович,

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ

**Веб-застосунок для генерації шкільного розкладу на основі еволюційного алгоритму**

Процес складання розкладу займає дуже багато часу та є доволі трудомісткою задачею. На даний момент проведено багато досліджень щодо процесу складання розкладу. Проте основна проблема полягає саме в реалізації програмного забезпечення, яке б враховувало специфіку обраного ринку та всі особливості даного процесу.

Метою даної роботи є розробка веб-застосування для генерації шкільного розкладу. В результаті було реалізовано веб-застосунок, який автоматизує процес складання шкільного розкладу, що в свою чергу зменшує кількість часу, який зазвичай витрачається на виконання даного завдання.

Еволюційний алгоритм. Відповідно до теорії еволюції Дарвіна, найпристосованіші особини в навколишньому середовищі виживають і передають свої риси наступним поколінням. Цей алгоритм є метаевристичним – це значить, що він здатний знайти прийнятне рішення проблеми серед багатьох рішень, але неспроможний гарантувати, що це рішення є оптимальним [1].

При описі еволюційного алгоритму використовують визначення, які є запозиченими з генетики:

* ген – атомарний елементи генотипу, зокрема, хромосоми;
* хромосома – впорядкована послідовність генів, яка являє собою певне рішення задачі;
* особина – сутність, яка включає в себе хромосому та значення пристосованості;
* популяція – множина особин, яка є набором рішень для певної проблеми.

Ефективність еволюційного алгоритму залежить від імплементації його головних процесів:

* визначення життєздатності особини – це процес, який дозволяє визначити пристосованість особини в чисельному вигляді. Для цього створюють функцію пристосованості (зазвичай є унікальною для кожної задачі);
* селекція – процес, який за певним принципом обирає деяку кількість особин з популяції, над якими будуть проводитися наступні операції;
* схрещування – процес, який завдяки змішуванню хромосом особин, обраних на етапі селекції, створює потомство. Основна мета цього етапу полягає в знаходженні нових рішень, які будуть містити найкращі елементи батьків;
* мутація – це процес, який піддає особину невеличкій випадковій зміні. Головна ідея являється в тому, аби алгоритм не попадав в локальний екстремум. Саме мутація дає можливість знайти нові специфічні рішення.
* заміна популяції – це процес, який замінює певних особин поточної популяції на ті, які були створені піл час схрещування та мутації.

Псевдокод алгоритму:

1) randomly create initial population

2) sort initial population

3) **while** termination criterion is not satisfied **do**

4) select parents from population

5) create offspring from parents using crossover

6) mutate offspring

7) replace the population with offspring

8) sort population

9) **end while**

10) return the best individual found during the evolution

Якщо поглянути на псевдокод алгоритму, який наведено вище, то відразу помітно, що він дещо відрізняється від традиційних. Основна відмінність полягає в рядках 2 та 8. В даній роботі було прийнято рішення реалізувати саме такий підхід, так як завдяки сортуванні популяції є можливість виконання етапів селекції та заміни популяції майже за константний час.

Дана розробка ставила перед собою досягнення наступних цілей:

* зменшення навантаження на людей, які займаються вирішенням проблеми складання розкладу;
* зменшення часу, який зазвичай витрачається на формування шкільного розкладу;
* підвищити ефективність учнів за рахунок створення зручного розкладу.

Для досягнення поставлених цілей були вирішені наступні задачі:

* імплементувати REST API для комунікації між серверною та клієнтською частинами веб-застосування;
* спроектувати базу даних для збереження необхідних даних;
* реалізувати алгоритм, який відповідає за генерацію розкладу.

Для написання серверної частини, було обрано сімейство фреймворку Spring [2], а саме: Spring Boot, Spring Data, Spring Web MVC та Spring Security. Фреймворки Spring мають велику спільноту розробників, гарну документацію та покривають майже весь функціонал сучасних веб-застосувань. Також було додано бібліотеку OpenApi, яка генерую документацію на основі Java анотацій. Було прийнято рішення імплементувати еволюційний алгоритм самостійно, аби мати повний контроль над усіма його компонентами. Для зберігання інформації обрано PostgreSQL як сховище даних, оскільки воно повністю відповідає вимогам та цілям розробки [3].

Для розробки клієнтської сторони було обрано фреймворк Angular, оскільки він вирішує велику кількість проблем самостійно, без допомоги сторонніх бібліотек та інструментів і забезпечує механізм модульності, систему директив, впровадження залежностей (DI) та інші [4].

На рисунку 1 представлено загальну архітектуру веб-застосування реалізованого в даній роботі.

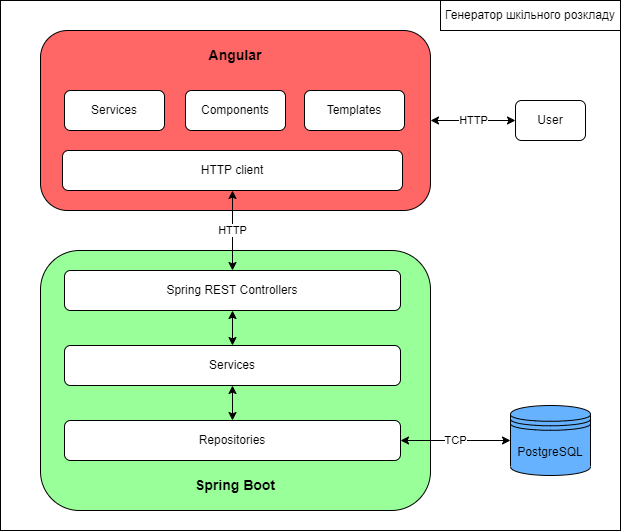


Рисунок 1. Загальна архітектура веб-застосування

**Література**

1. Evolutionary algorithms [Електронний ресурс] – Режим доступу до ресурсу: <https://www.researchgate.net/publication/261842296_Evolutionary_Algorithms>.
2. Spring Framework Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.spring.io/spring-framework/reference/>.
3. PostgreSQL: The world’s most advanced open source database [Електронний ресурс] – Режим доступу до ресурсу: <https://www.postgresql.org/>.
4. Angular [Електронний ресурс] – Режим доступу до ресурс: <https://angular.io/>.