



**ŽILINSKÁ UNIVERZITA V ŽILINE**

**Fakulta riadenia  
a informatiky**

Semestrálna práca z predmetu *vývoj  
aplikácií pre mobilné zariadenia*

**ALGORITHM VISUALIZER**

**Vypracoval:** Filip Dávid

**Študijná skupina:** 5ZYR22

**Akademický rok:** 2024/2025

V Žiline dňa 8.6.2025



## Obsah

Obsah .....	1
Úvod .....	1
Prehľad podobných aplikácií .....	2
Analýza navrhovanej aplikácie .....	4
Návrh architektúry aplikácie .....	5
Návrh vzhľadu obrazoviek .....	6
Zoznam zdrojov .....	11

## Použité obrázky

Obrázok 1 Visual Sort Algorithms Quick sort .....	2
Obrázok 2 Sorting Algorithms Visualizer Bubble sort .....	3
Obrázok 3 Sorting Algorithms – Code Viewer Bubble sort .....	4
Obrázok 4 Use case diagram .....	5
Obrázok 5 Hlavná obrazovka .....	6
Obrázok 6 Obrazovka triediacich algoritmov .....	7
Obrázok 7 Obrazovka vyhľadávacích algoritmov .....	8
Obrázok 8 Reálna obrazovka s krokovaním, rýchlosťou a reset tlačidlom .....	9
Obrázok 9 Funkcionalita krokovanie – obrazovky pre triediace a vyhľadávacie algoritmy .....	10
Obrázok 10 Obrazovka pre vkladanie nezoradeného poľa do databázy .....	10

## Úvod

Algorithm Visualizer je interaktívna aplikácia, ktorá bola navrhnutá s cieľom pomôcť používateľom vizualizovať fungovanie triediacich a vyhľadávacích algoritmov. Aplikácia je určená predovšetkým pre študentov informatiky, ktorí sa chcú hlbšie ponoriť do fungovania základných algoritmov a ich efektívnosti. Cieľom aplikácie je nielen ukázať, ako tieto algoritmy pracujú, ale aj poskytnúť vizuálny a interaktívny spôsob učenia, ktorý umožní používateľom lepšie pochopiť, ako jednotlivé algoritmy manipulujú s dátami krok za krokom. Prostredníctvom animácií a dynamického vykonávania algoritmov sa používateľ môže pozrieť na každý krok, čo zjednodušuje pochopenie ich vnútornej logiky.

Aplikácia ponúka dve hlavné kategórie: triediace algoritmy a vyhľadávacie algoritmy, pričom v každej kategórii sú dostupné rôzne populárne algoritmy, ako napríklad Bubble Sort, Quick Sort, Binary Search, a ďalšie. Používateľ si môže vybrať algoritmus, sledovať jeho vykonávanie a analyzovať časovú a výpočtovú zložitosť. Tento prístup nielen uľahčuje učenie, ale aj poskytuje praktické skúsenosti v oblasti algoritmiky, čo je neoceniteľné pre každého, kto sa venuje informatike a programovaniu.

## Prehľad podobných aplikácií

### 1. Visual Sort Algorithms

Funkcie:

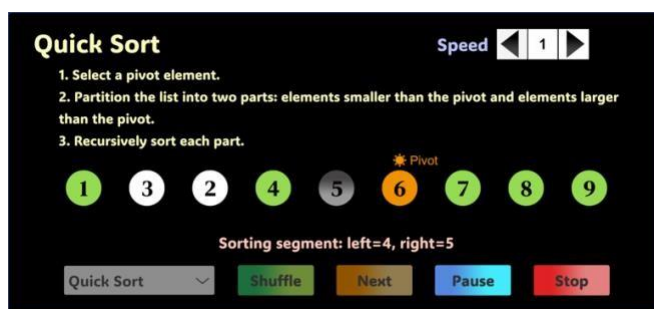
Vizualizácia viacerých triediacich algoritmov (napr. Bubble Sort, Merge Sort, Quick Sort). Audiovizuálna prezentácia triedenia s možnosťou meniť počet prvkov. Animácia triedenia v reálnom čase.

Výhody:

- Pútavá animácia.
- Jednoduché ovládanie.

Nevýhody:

- Chýba podpora vyhľadávacích algoritmov.
- Zastaralý dizajn
- Vizualizácia len cez čísla



Obrázok 1 Visual Sort Algorithms Quick sort

[1]

### 2. Sorting Algorithms Visualizer

Funkcie:

Vizualizácia viacerých triediacich algoritmov (napr. Bubble Sort, Merge Sort, Quick Sort). Generovanie náhodných čísel alebo ručný vstup. Zobrazovanie počtu porovnaní a výmen. Animácia triedenia v reálnom čase. Časovač a posuvník rýchlosti triedenia.

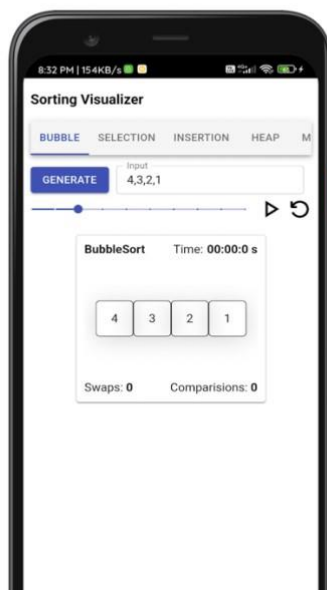
Výhody:

- Väčšia kontrola nad vizualizáciou (rýchlosť, vstupy). - Užitočné pre sledovanie výkonu algoritmov.

Nevýhody:

- Chýba podpor Zamerané len na triediace algoritmy.
- Nie je určené na výučbu – chýbajú popisy, vysvetlenia. - Neobsahuje vyhľadávacie algoritmy.

## **SORT**



Obrázok 2 Sorting Algorithms Visualizer Bubble sort

[2]

### 3. Sorting Algorithms – Code Viewer

Funkcie:

Zobrazuje popis algoritmov a ich implementácie vo viacerých jazykoch.

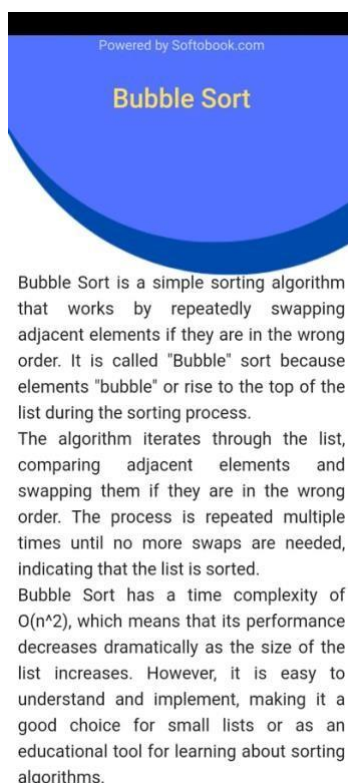
Žiadna vizualizácia – len text + kód.

Výhody:

- Skvelé ako referenčný nástroj pre programátorov. - Pokrýva širokú škálu programovacích jazykov.

Nevýhody:

- Žiadna animácia ani interaktívna zložka.
- Nevhodné pre vizuálne učenie alebo študentov, ktorí preferujú dynamické znázornenie.



Obrázok 3 Sorting Algorithms – Code Viewer Bubble sort

[3]

## Analýza navrhovanej aplikácie

V Aplikácia **Algorithm Visualizer** je navrhnutá pre záujemcov o informatiku, ktorí si chcú vizuálne osvojiť fungovanie triediacich a vyhľadávacích algoritmov. Používateľ si môže vybrať konkrétny algoritmus, sledovať jeho priebeh prostredníctvom animácie a analyzovať jeho efektívnosť. Oproti existujúcim riešeniam ponúka niekoľko praktických vylepšení, ako je interaktívna vizualizácia, možnosť prepínať medzi algoritmami a sledovať ich porovnanie.

Funkcie aplikácie:

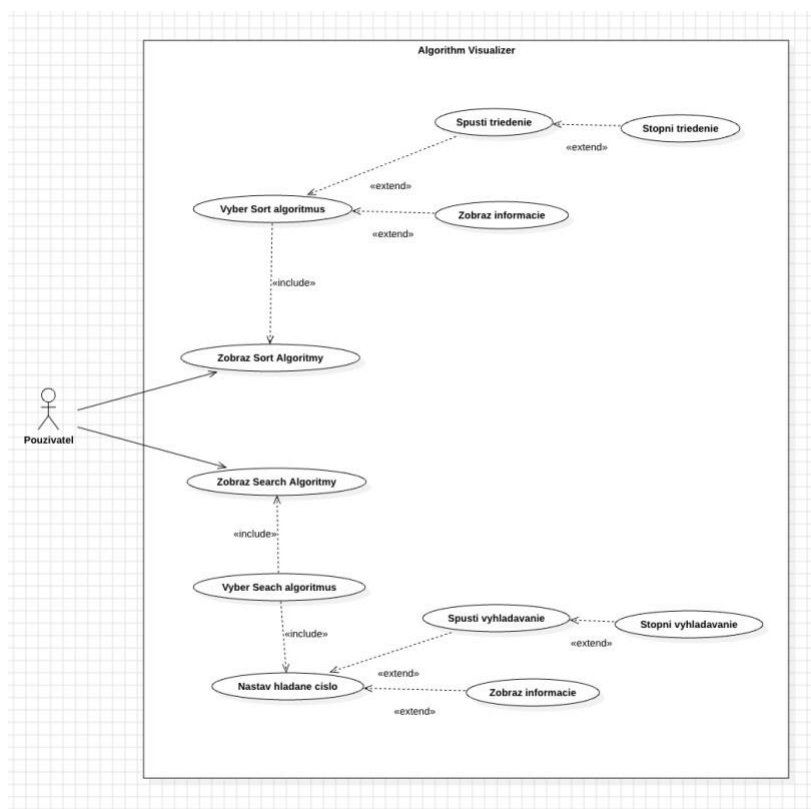
Výber algoritmu – Používateľ si môže vybrať medzi rôznymi triediacimi a vyhľadávacími algoritmami. Aplikácia umožňuje vizualizáciu rôznych typov algoritmov, ako sú Bubble Sort, Quick Sort, Linear Search, Binary Search a ďalšie.

Vizualizácia algoritmu – Po výbere algoritmu aplikácia zobrazuje jednotlivé kroky jeho vykonávania, pričom animuje proces tak, aby používateľ mohol vidieť, ako sa hodnoty menia počas vykonávania algoritmu.

Analýza efektivity – Aplikácia poskytuje informácie o efektívnosti algoritmu, ako napríklad jeho časovú a priestorovú zložitosť (napr.  $O(n)$ ,  $O(n^2)$ ).

Interaktívne ovládanie – Používateľ môže pauzovať, preskočiť na ďalší krok, alebo spustiť algoritmus znovu s inými dátami. Tento interaktívny prístup pomáha pri lepšom pochopení fungovania algoritmov.

Porovnanie algoritmov – Používateľ má možnosť porovnať rôzne algoritmy na rovnakých dátach a vizuálne sledovať, ktorý algoritmus je efektívnejší v konkrétnych podmienkach.



Obrázok 4 Use case diagram

## Návrh architektúry aplikácie

Aplikácia Algorithm Visualizer je navrhnutá na vizualizáciu triediacich a vyhľadávacích algoritmov. Cieľom aplikácie je poskytovať používateľom interaktívny nástroj na vizualizáciu fungovania algoritmov, čím umožňuje lepšie pochopenie ich logiky a efektivity. Používateľ si môže vybrať algoritmus, sledovať jeho priebeh cez animácie a analyzovať jeho efektívnosť na základe zobrazených informácií.

**Aplikácia je postavená na architektúre Model-View-ViewModel (MVVM), kde:**

**Model:** Reprezentuje logiku aplikácie, konkrétne triediace a vyhľadávacie algoritmy. Tieto algoritmy sú implementované ako samostatné moduly, ktoré nie sú závislé na grafickom rozhraní. Model sa stará o samotný výpočet algoritmov a vracia výsledky.

**View:** Zodpovedá za grafické rozhranie aplikácie. Zobrazuje pole, krok algoritmu, informácie o aktuálnom stave a všetky interaktívne prvky, ako sú tlačidlá na spustenie a zastavenie algoritmov.

**ViewModel:** Prepojuje Model a View. Obsahuje stavové premenné, ktoré uchovávajú aktuálny stav aplikácie, ako je vybraný algoritmus, aktuálne pole, aktuálny krok v algoritme, a podobne. ViewModel tiež komunikuje s Modelom, aby vykonal výpočty, a aktualizuje View podľa zmeny stavu.

### Dátový model

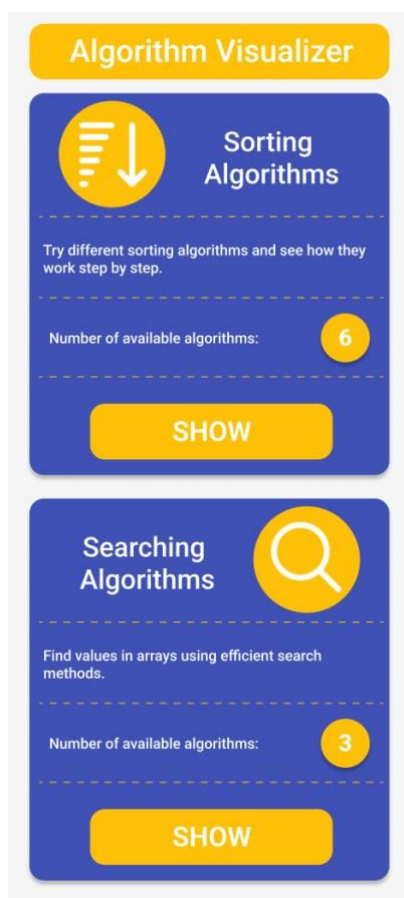
Aplikácia pracuje s rôznymi dátami, ktoré reprezentujú pole čísel počas vykonávania algoritmov. Tento zoznam čísel je dynamicky aktualizovaný v reálnom čase, aby sa zobrazovala vizualizácia postupu

algoritmu. Dáta sa udržiavajú v pamäti a sú manipulované algoritmami, ktoré ich triedia alebo vyhľadávajú.

### Funkčnosť aplikácie

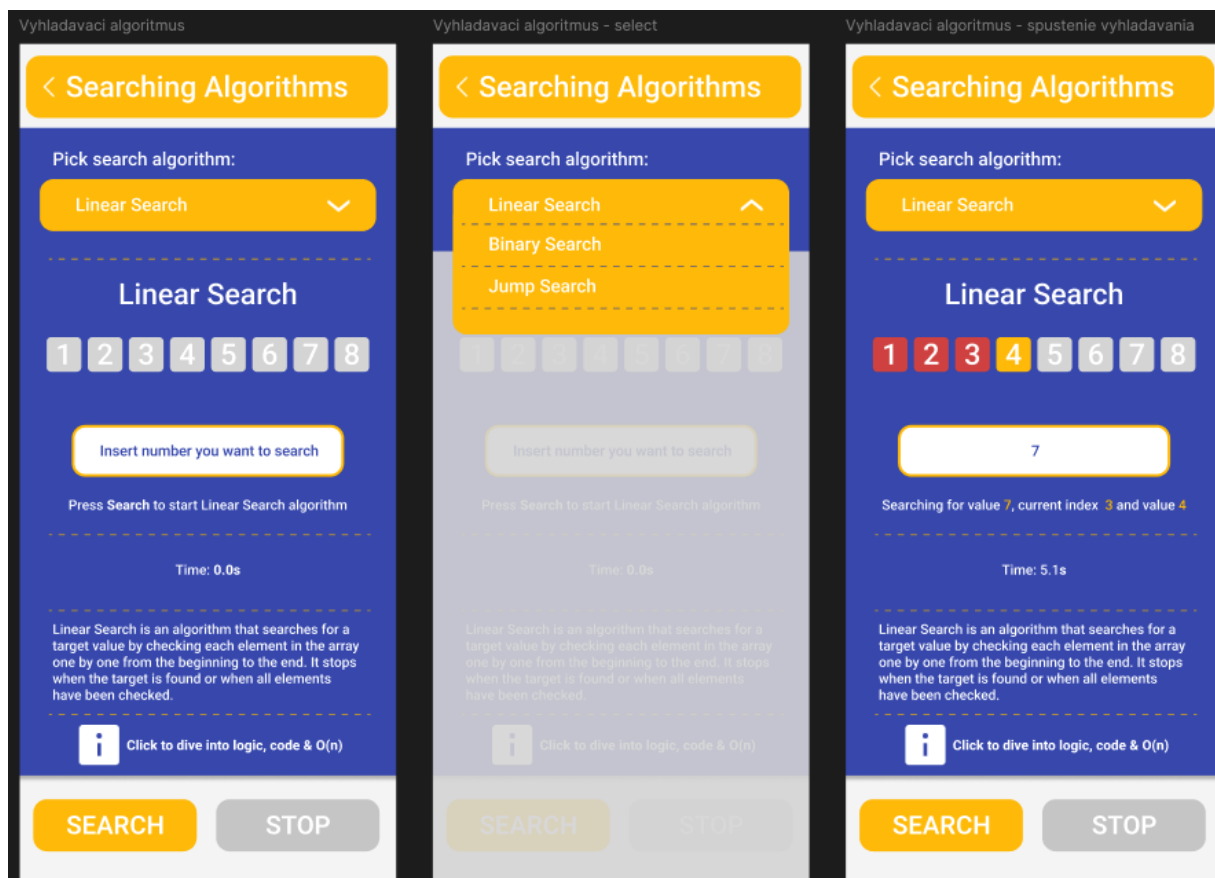
1. **Výber algoritmu:** Používateľ na začiatku vyberie medzi triediacimi a vyhľadávacími algoritmami. Tým sa zmení obsah aplikácie a načítajú sa možnosti konkrétnych algoritmov v danej kategórii.
2. **Animácia algoritmu:** Po spustení algoritmu sa zobrazuje animácia jeho krokov, ktorá ukazuje, ako algoritmus manipuluje s poľom. Každý krok je vizualizovaný, aby používateľ jasne videl, čo sa deje v reálnom čase.
3. **Zastavenie algoritmu:** Používateľ má možnosť zastaviť vykonávanie algoritmu kedykoľvek.
4. **Zobrazenie informácií:** Počas vykonávania algoritmu je zobrazené stručné vysvetlenie toho, čo algoritmus robí, ako aj časové a výpočtové zložitosti. Po kliknutí na "Info" ikonu sa zobrazí podrobné vysvetlenie o algoritme vrátane kódu.

### Návrh vzhľadu obrazoviek



Obrázok 5 Hlavná obrazovka

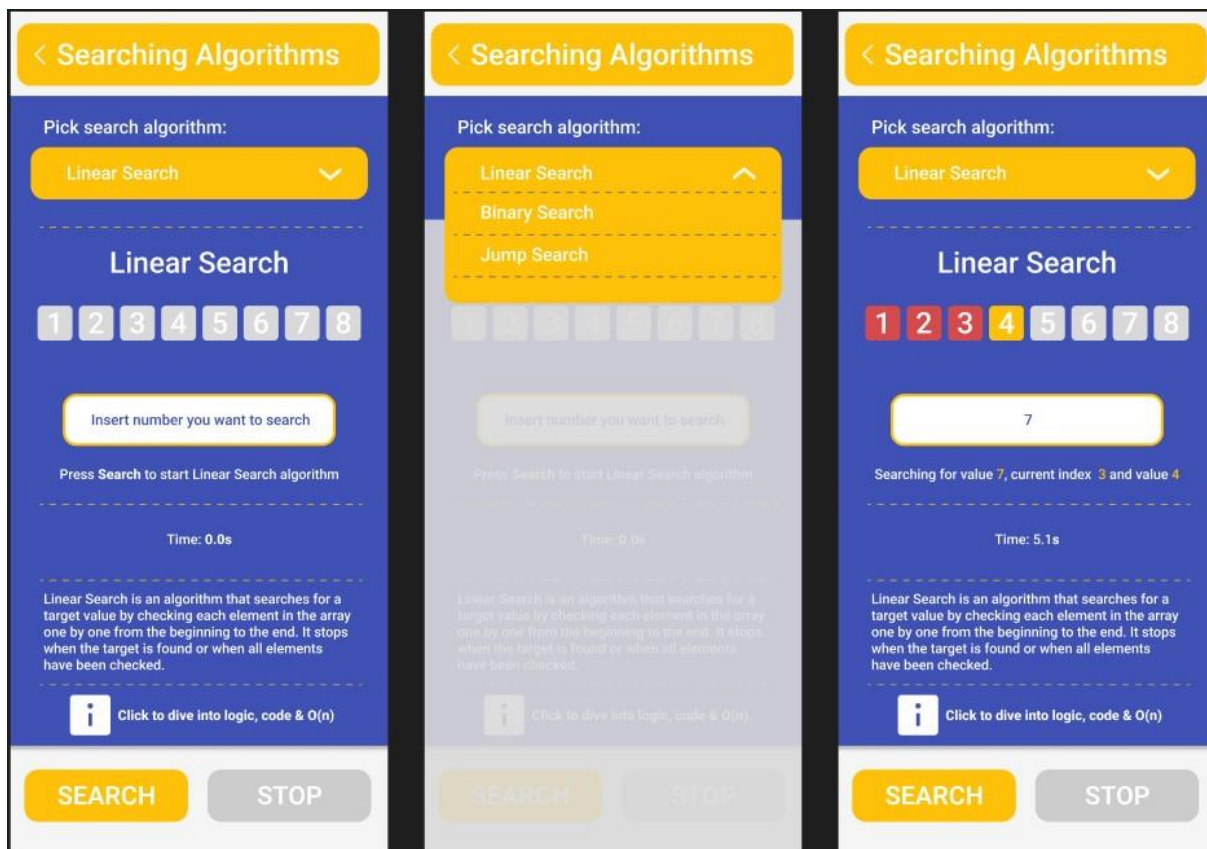
Na hlavnej obrazovke sa nachádzajú dve karty, ktoré reprezentujú kategórie algoritmov: triediace algoritmy a vyhľadávacie algoritmy. Každá karta zobrazuje stručnú informáciu o tom, čo používateľ môže očakávať, ak si ju rozklikne. Zobrazuje sa aj počet dostupných algoritmov v danej kategórii.



Obrázok 6 Obrazovka triediacich algoritmov

Obrazovka triediacich algoritmov umožňuje používateľovi vybrať jeden z dostupných triediacich algoritmov. Zobrazuje stĺpcový graf, ktorý graficky reprezentuje pole počas procesu triedenia. Pod grafom sa nachádza informácia o aktuálnych krokoch podľa vybraného algoritmu (napr. porovnávanie prvkov pri použití Bubble Sort). Ďalej sa zobrazuje čas od spustenia triedenia a stručná informácia o algoritme. Po kliknutí na „Info“ ikonu sa zobrazí podrobné informácie, ako napríklad časová a výpočtová zložitosť, kód a ďalšie detaily. Na spodnej časti obrazovky sú tlačidlá na spustenie a zastavenie triedenia.





Obrázok 7 Obrazovka vyhľadávacích algoritmov

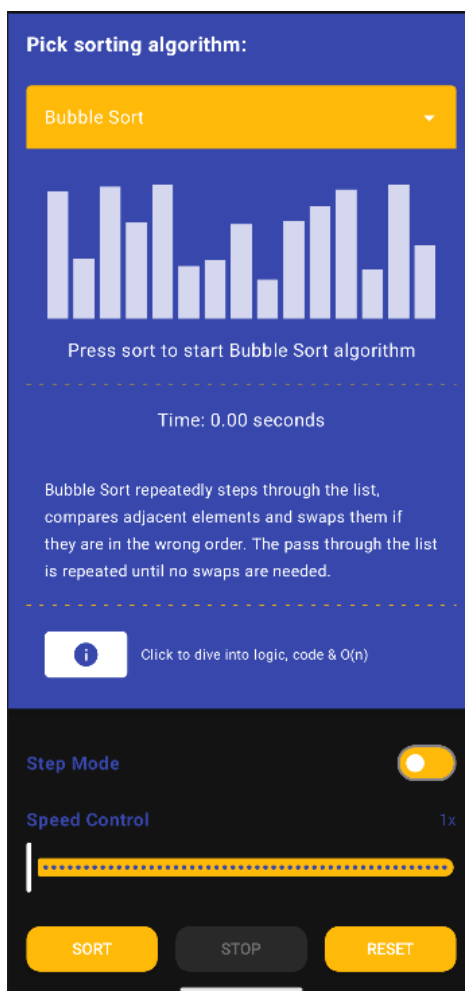
Obrazovka vyhľadávacích algoritmov umožňuje používateľovi vybrať jeden z dostupných vyhľadávacích algoritmov. Zobrazuje čísla ako prvky v poli, ktoré sú zobrazené ako jednotlivé bunky. Pod bunkami sa nachádza textové pole na zadanie hľadaného prvku. Ďalej sa zobrazuje informácia o aktuálnych krokoch podľa vybraného algoritmu, ako je hľadaná hodnota, aktuálny index a hodnota na tomto indexe. Okrem toho sa zobrazuje čas od začiatku vyhľadávania a stručná informácia o vybranom algoritme. Po kliknutí na ikonu „Info“ sa zobrazí podrobné informácie, ako napríklad časová a výpočtová zložitosť, kód a ďalšie detaily. Na spodnej časti obrazovky sú tlačidlá na spustenie a zastavenie vyhľadávania.

## Skutočný návrh riešenia problému

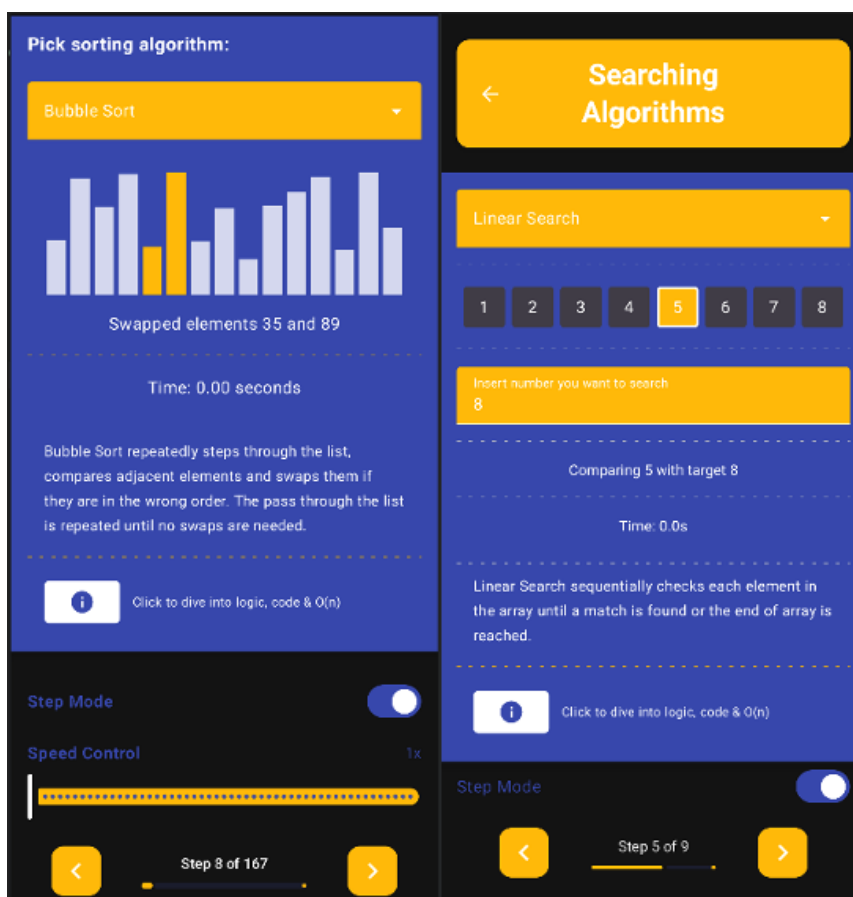
Do aplikácie boli pridané ďalšie funkcionality či už potrebné ako napríklad resetovanie poľa po zoradení, pridanie možnosti krokovania pre lepšie pochopenie jednotlivých algoritmov a to či už triediacich alebo vyhľadávacích. Taktiež sa pridala možnosť zrýchliť triedenie až na rýchlosť 50x čo umožňuje v pomerne krátkej dobe zoradiť aj 40+ prvkové polia.

Pridala sa aj práca s databázou firestore od firebase odkiaľ sa načítavajú nezoradené polia. Taktiež je možné si definovať vlastné pomocou určenia veľkosti poľa, vygenerovania náhodných hodnôt a nahrať do firestore.

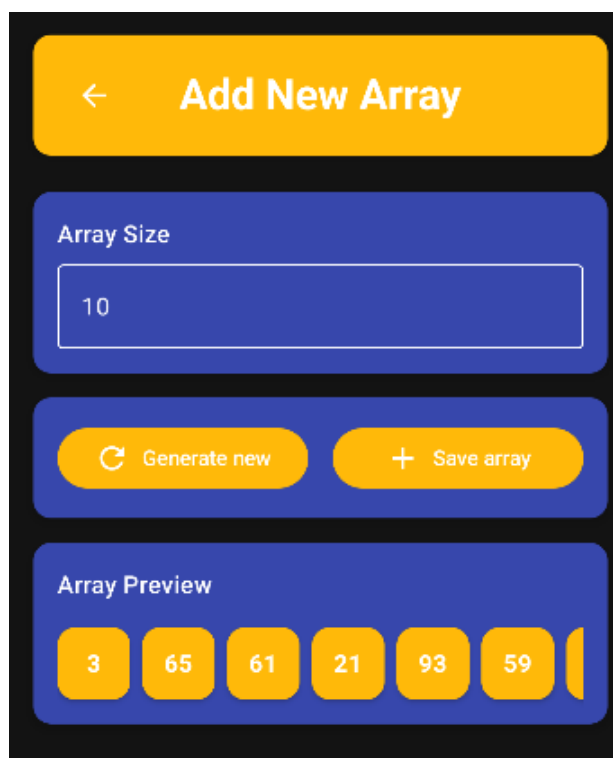
Nezoradené pole sa náhodne načíta z databázy. Ak nastane nejaký problém alebo užívateľ nemá prístup k internetu použije sa náhradné pole.



Obrázok 8 Reálna obrazovka s krokovaním, rýchlosťou a reset tlačidlom



Obrázok 9 Funkcionalita krokovanie – obrazovky pre triediace a vyhľadávacie algoritmy



Obrázok 10 Obrazovka pre vkladanie nezoradeného poľa do databázy



## Zoznam zdrojov

1. Sorting Algorithms Visualizer. Online. Dostupné na:  
<https://play.google.com/store/apps/details?id=com.microgames.apps.sorts>
2. Sorting Algorithms Visualizer. Online. Dostupné na:  
[https://play.google.com/store/apps/details?id=com.fazil.sorting\\_visualizer&hl=en](https://play.google.com/store/apps/details?id=com.fazil.sorting_visualizer&hl=en)
3. Sorting Algorithms – Code Viewer Online. Dostupné na:  
<https://play.google.com/store/apps/details?id=ujapps.softobook.com.softo.apps.sortingalgorithms>