

## Specifikacija funkcionalnosti

### 1. Naziv funkcionalnosti: Računanje popusta prilikom rentanja

**Opis funkcionalnosti:** Korisnik prilikom rentanja određenog alata ima mogućnost unijeti kod, pomoću kojeg će, ukoliko je kod validan, dobiti popust na ukupnu cijenu rentanja. Pored toga što kod mora biti validan, trenutak u kojem korisnik vrši rentanje se mora nalaziti u dozvoljenom intervalu trajanja popusta (nakon početka popusta i poslije završetka popusta).

### 2. Naziv funkcionalnosti: Lista predloženih alata

**Opis funkcionalnosti:** Korisnik će prilikom ulaska na aplikaciju, imati prikazanu listu predloženih alata koja je formirana na osnovu njegove historije rentanja (ukoliko je korisnik do tada rentao alate iz samo jedne kategorije, cijela predložena lista će biti iz alata te kategorije, analogno tome, ukoliko je korisnik u svojoj historiji rentao 50% jednu kategoriju i 50% drugu kategoriju, isti omjer će biti prenešen na kategorije predloženih alata).

## Implementirane metode

```
37 | 7 references | Please sign-in to New Relic CodeStream to see Code Level Metrics | 6/6 passing
38 | public double CalculateDiscount(Tool tool, int code, List<Discount> discounts) {
39 |     double toolPrice = tool.ToolType.Price;
40 |     double discountPrice;
41 |     for (int i = 0; i < discounts.Count; i++) {
42 |         if (code == discounts[i].Id) {
43 |             if (DateTime.Now < discounts[i].StartOfDiscount || DateTime.Now > discounts[i].EndOfDiscount) {
44 |                 throw new InvalidOperationException();
45 |             } else {
46 |                 discountPrice = toolPrice * (1 - discounts[i].Percent / 100);
47 |                 return discountPrice;
48 |             }
49 |         }
50 |     }
51 |     throw new ArgumentException();
    }
```

```
57 | 13 references | Please sign-in to New Relic CodeStream to see Code Level Metrics | 12/12 passing
58 | public List<ToolType> SuggestedTools(List<ToolType> history, List<ToolType> tools) {
59 |     if (history.Count == 0)
60 |         throw new ArgumentException();
61 |
62 |     List<ToolType> result = new List<ToolType>();
63 |     Dictionary<Category, int> map = new Dictionary<Category, int>();
64 |     for (int i = 0; i < history.Count; i++) {
65 |         if (map.ContainsKey(history[i].Category)) {
66 |             map[history[i].Category]++;
67 |         } else {
68 |             map[history[i].Category] = 1;
69 |         }
    }
```

## Uloge i aktivnosti članova

- **Emir Agović** – definiranje testova za funkcionalnosti, implementiranje metoda *CalculateDiscount* i *SuggestedTools*
- **Ahmed Ljubunčić** – integrisanje dodanih metoda u ostatak programa (npr. frontend)