



Univerzitet u Sarajevu
Elektrotehnički fakultet
Sarajevo

TASK 4

Sistem za upravljanje zalihama koktel bara **TDD**



Predmet: Verifikacija i validacija softvera

Odsjek: Računarstvo i informatika

Odgovorni nastavnik: R. prof. dr. Dženana Đonko dipl.el.ing

Odgovorni asistent: Mr. dipl. ing. Neira Novalić

ČLANOVI TIMA:

- Amina Hromić
- Emina Efendić
- Hana Mahmutović
- Tajra Selimović

Sarajevo, decembar 2023.

Sadržaj

Funkcionalnosti.....	3
Član tima 1: Amina Hromić.....	4
Član tima 2: Emina Efendić.....	6
Član tima 3: Hana Mahmutović.....	7
Član tima 4: Tajra Selimović.....	8
Pokrivenost koda testovima.....	9

Funkcionalnosti

Dvije nove funkcionalnosti koje smo odlučile dodati kako bismo riješile ovaj task su:

1. Prvovjera da li su sva pića u ponuđenoj listi pića alkoholna - ovo smo realizirale pomoću metode `AreAllDrinksAlcoholic` koja kao jedini parametar prima listu pića. Za svako piće u listi pozivamo funkciju `isAlcoholic` koja kao parametar prima piće (tip `Drink`) i provjerava da li je bar jedan od njegovih sastojaka alkoholni.
2. Provjera da li je nastupio "Happy Hour" (između 20:00 h i 22:00 h) te ukoliko jeste, smanjivanje cijena pića za 20%. Ove funkcionalnosti su realizovane respektivno pomoću metode `CheckHappyHourStatus` koja prima `DateTime` atribut te vraća `true` ukoliko je "Happy Hour" nastupio, a `false` ukoliko nije te metode `ApplyHappyHourDiscount` koja prima listu pića i `DateTime` atribut. Kao rezultat vraća listu pića sa sniženim cijenama ukoliko je "Happy Hour" nastupio.

Član tima 1: Amina Hromić

Napisani su testovi za testiranje druge funkcionalnosti (red faza). U nastavku je prikazan izgled testova u green fazi.

```
291 // Test driven development
292 [TestMethod]
293 | 0 references | we ran into an exception loading metrics for this method - please contact support
294 public void CheckHappyHourStatus_StartOfHappyHour_True()
295 {
296     DateTime dateTime = new DateTime(2023, 12, 31, 20, 0, 0); //31.12.2023. u 20:00
297     var result = controller.CheckHappyHourStatus(dateTime);
298     Assert.IsTrue(result);
299 }
300
301 [TestMethod]
302 | 0 references | we ran into an exception loading metrics for this method - please contact support
303 public void CheckHappyHourStatus_MiddleOfHappyHour_True()
304 {
305     DateTime dateTime = new DateTime(2023, 12, 31, 21, 30, 0); //31.12.2023. u 21:30
306     var result = controller.CheckHappyHourStatus(dateTime);
307     Assert.IsTrue(result);
308 }
309
310 [TestMethod]
311 | 0 references | we ran into an exception loading metrics for this method - please contact support
312 public void CheckHappyHourStatus_EndOfHappyHour_True()
313 {
314     DateTime dateTime = new DateTime(2023, 12, 31, 22, 0, 0); //31.12.2023. u 22:00
315     var result = controller.CheckHappyHourStatus(dateTime);
316     Assert.IsTrue(result);
317 }
318
319 [TestMethod]
320 | 0 references | we ran into an exception loading metrics for this method - please contact support
321 public void CheckHappyHourStatus_BeforeHappyHour_False()
322 {
323     DateTime dateTime = new DateTime(2023, 12, 31, 19, 0, 0); //31.12.2023. u 19:00
324     var result = controller.CheckHappyHourStatus(dateTime);
325     Assert.IsFalse(result);
326 }
327
328 [TestMethod]
329 | 0 references | we ran into an exception loading metrics for this method - please contact support
330 public void CheckHappyHourStatus_AfterHappyHour_False()
331 {
332     DateTime dateTime = new DateTime(2023, 12, 31, 23, 0, 0); //31.12.2023. u 19:00
333     var result = controller.CheckHappyHourStatus(dateTime);
334     Assert.IsFalse(result);
335 }
```

```

331
332 [TestMethod]
333 | 0 references | we ran into an exception loading metrics for this method - please contact support
334 public void ApplyHappyHourDiscount_HappyHour_ReturnsDiscountedDrinks()
335 {
336     DateTime dateTime = new DateTime(2023, 12, 31, 21, 0, 0); //31.12.2023. u 21:00
337     var drinks = new List<Drink>{ new Drink(1, "Grapefruit Spritz", 4.99),
338     new Drink(2, "Orange Blossom", 2.99), new Drink(3, "Kiwi Kiss", 6.99)
339     };
340     var drinksExpected = new List<Drink>{ new Drink(1, "Grapefruit Spritz", Math.Round(4.99 * 0.8, 1)),
341     new Drink(2, "Orange Blossom", Math.Round(2.99 * 0.8, 1)), new Drink(3, "Kiwi Kiss", Math.Round(6.99 * 0.8, 1))
342     };
343     var result = controller.ApplyHappyHourDiscount(drinks, dateTime);
344
345     Assert.AreEqual(drinksExpected[0].price, result[0].price);
346     Assert.AreEqual(drinksExpected[1].price, result[1].price);
347     Assert.AreEqual(drinksExpected[2].price, result[2].price);
348 }

```

```

349
350 [TestMethod]
351 | 0 references | we ran into an exception loading metrics for this method - please contact support
352 public void ApplyHappyHourDiscount_HappyHour_ReturnsFullPriceDrinks()
353 {
354     DateTime dateTime = new DateTime(2023, 12, 31, 10, 0, 0); //31.12.2023. u 10:00
355     var drinks = new List<Drink>{ new Drink(1, "Grapefruit Spritz", 4.99),
356     new Drink(2, "Orange Blossom", 2.99), new Drink(3, "Kiwi Kiss", 6.99)
357     };
358     var result = controller.ApplyHappyHourDiscount(drinks, dateTime);
359
360     Assert.AreEqual(drinks[0].price, result[0].price);
361     Assert.AreEqual(drinks[1].price, result[1].price);
362     Assert.AreEqual(drinks[2].price, result[2].price);
363 }

```

Član tima 2: Emina Efendić

Napisan jedan test za prvu funkcionalnost koji pokriva slučaj kada se u listi pića nalaze samo alkoholna pića.

```
342 [TestMethod]
343 | 0 references | we ran into an exception loading metrics for this method - please contact support
344 public void AreAllDrinksAlcoholic_AllDrinksAlcoholic_ReturnsTrue()
345 {
346     var drinks = new List<Drink>
347     {
348         new Drink(2, "Boody Mary", 6.49),
349         new Drink(3, "Blue Lagoon", 7.99),
350         new Drink(4, "Blue Lemonade", 7.88)
351     };
352     var mockDbContext = new Mock<IApplicationDbContext>();
353
354     controller = new AdminPanelController(mockDbContext.Object);
355
356     var drinkIngredientsData = new List<DrinkIngredient>
357     {
358         new DrinkIngredient { id = 1, idDrink = 2, idIngredient = 21 },
359         new DrinkIngredient { id = 2, idDrink = 3, idIngredient = 27 },
360         new DrinkIngredient { id = 3, idDrink = 4, idIngredient = 29 }
361     };
362
363     mockDbContext.Setup(c => c.DrinkIngredients).Returns(MockDbSet(drinkIngredientsData));
364     var result = controller.AreAllDrinksAlcoholic(drinks);
365
366     Assert.IsTrue(result);
367 }
```

Urađena implementacija metode isAlcoholic koja prima piće kao parametar.

```
313 1 reference | we ran into an exception loading metrics for this method - please contact support
314 public bool isAlcoholic(Drink drink)
315 {
316     List<int> alcoholicIngredientId = new List<int> { 21, 27, 29, 33, 42, 54 };
317
318     var drinkIngredients = _context.DrinkIngredients
319     .Where(di => di.idDrink == drink.id && alcoholicIngredientId.Contains(di.idIngredient))
320     .ToList();
321
322     return drinkIngredients.Any();
323 }
```

Član tima 3: Hana Mahmutović

Urađena implementacija metoda `CheckHappyHourStatus` i `ApplyHappyHourDiscount`.

```
320 }
321 //Test driven development
322 7 references | we ran into an exception loading metrics for this method - please contact support | 5/5 passing
323 public bool CheckHappyHourStatus(DateTime dateTime)
324 {
325     var happyHourStartTime = new TimeSpan(20, 0, 0); // 8 PM
326     var happyHourEndTime = new TimeSpan(22, 0, 0); // 10 PM
327
328     return dateTime.TimeOfDay >= happyHourStartTime && dateTime.TimeOfDay <= happyHourEndTime;
329 }
330
331 3 references | we ran into an exception loading metrics for this method - please contact support | 2/2 passing
332 public List<Drink> ApplyHappyHourDiscount(List<Drink> drinks, DateTime dateTime)
333 {
334     if (CheckHappyHourStatus(dateTime))
335     {
336         foreach (var drink in drinks)
337         {
338             drink.price = Math.Round(drink.price * 0.8, 1); // 20% popusta
339         }
340     }
341     return drinks;
342 }
```

Član tima 4: Tajra Selimović

Napisan jedan test za prvu funkcionalnost koji pokriva slučaj kada se u listi pića ne nalaze samo alkoholna pića.

```
265 [TestMethod]
266 | 0 references | we ran into an exception loading metrics for this method - please contact support
267 public void AreAllDrinksAlcoholic_AllDrinksDoNotContainAlcoholicIngredients_ReturnsFalse()
268 {
269     // Arrange
270     var drinkIngredients = new List<DrinkIngredient>
271     {
272         new DrinkIngredient(1, 1, 21),
273         new DrinkIngredient(2, 1, 23),
274         new DrinkIngredient(3, 1, 29),
275     };
276     var mockDbContext = new Mock<IApplicationDbContext>();
277     mockDbContext.Setup(c => c.DrinkIngredients).Returns(MockDbSet(drinkIngredients));
278
279     controller = new AdminPanelController(mockDbContext.Object);
280
281     var drinks = new List<Drink>
282     {
283         new Drink { id = 1, name = "AlcoholicDrink1", price = 8.99 },
284         new Drink { id = 2, name = "AlcoholicDrink2", price = 7.99 },
285     };
286
287     // Act
288     var result = controller.AreAllDrinksAlcoholic(drinks);
289
290     // Assert
291     Assert.IsFalse(result);
292 }
```

Urađena implementacija metode AreAllDrinksAlcoholic koja prima listu pića kao parametar.

```
324 2 references | we ran into an exception loading metrics for this method - please contact support | 2/2 passing
325 public bool AreAllDrinksAlcoholic(List<Drink> drinks)
326 {
327     foreach (var drink in drinks)
328     {
329         Console.WriteLine("drink 1");
330         if (!isAlcoholic(drink))
331         {
332             Console.WriteLine("nije ok");
333             return false;
334         }
335     }
336     Console.WriteLine("ok");
337     return true;
338 }
```


Pokrivenost koda testovima

Name	Covered	Uncovered	Coverable	Total	Line coverage	Covered	Total	Branch coverage
UnitTests	846	20	866	1441	97.6%	21	22	95.4%
VVSProject	246	95	341	596	72.1%	76	94	80.8%
SmartCafe.Controllers.AdminPanelController	214	15	229	338	93.4%	76	92	82.6%