Merima Durić 19025

Metoda koja se testira:

```
public List<SelectListItem> GetFullNames(List<StudentCourse> owo)
{
    List<SelectListItem> Students = new List<SelectListItem>();

    foreach (StudentCourse item in owo)
    {
        if (item.Student != null)
        {
            Students.Add(new SelectListItem() { Text = $"{item.Student.FirstName} {item.Student.LastName}", Value = item.ID.ToString() });
        }
    }

    return Students;

}
```

Line i branch coverage :

Jedan test pokriva oboje

```
[TestMethod]
| 0 references | Please sign-in to New Relic CodeStream to see Code Level Metrics
public async Task Test1()
{
    // Arrange
    var student = new Student { Id = 1, FirstName = "Jane", LastName = "Doe", UserName = "janeDOe", Email = "jane@example.com", Index = 12345, Department = "RI", Year = 3 };
    var studentCourse = new List<StudentCourse>

new StudentCourse { ID = 99, CourseID = 1, StudentID = 1,Student = student,  Points = 75, Grade = 8 },
new StudentCourse { ID = 99, CourseID = 1, StudentID = 1,Student = student,  Points = 75, Grade = 8 },
new StudentCourse { ID = 99, CourseID = 1, StudentID = 1,Student = student,  Points = 75, Grade = 8 },


    var controller = new StudentExamController(_mockContext.Object, _mockUserManager.Object);


    //Act
    var results = controller.GetFullNames(studentCourse);

    //Assert
    Assert.AreEqual(3, results.Count);
}
```

Conditional coverage:

```
[TestMethod]
| 0 references | Please sign-in to New Relic CodeStream to see Code Level Metrics
public async Task Test1()
{
    // Arrange
    var student = new Student { Id = 1, FirstName = "Jane", LastName = "Doe", UserName = "janeDOe", Email = "jane@example.com", Index = 12345, Department = "RI", Year = 3 };
    var studentCourse = new List<StudentCourse>

new StudentCourse { ID = 99, CourseID = 1, StudentID = 1,Student = student,  Points = 75, Grade = 8 },
new StudentCourse { ID = 99, CourseID = 1, StudentID = 1,Student = student,  Points = 75, Grade = 8 },
new StudentCourse { ID = 99, CourseID = 1, StudentID = 1,Student = student,  Points = 75, Grade = 8 },


    var controller = new StudentExamController(_mockContext.Object, _mockUserManager.Object);


    //Act
    var results = controller.GetFullNames(studentCourse);

    //Assert
    Assert.AreEqual(3, results.Count);
}
```

```
    [TestMethod]
    | 0 references | Please sign-in to New Relic CodeStream to see Code Level Metrics
    public async Task Test2()
    {
        // Arrange
        var student = new Student { Id = 1, FirstName = "Jane", LastName = "Doe", UserName = "janeDOe", Email = "jane@example.com", Index = 12345, Department = "RI", Year = 3 };
        var studentCourse = new List<StudentCourse>
{
    new StudentCourse { ID = 99, CourseID = 1, StudentID = 1,  Points = 75, Grade = 8 }
};


        var controller = new StudentExamController(_mockContext.Object, _mockUserManager.Object);


        //Act
        var results = controller.GetFullNames(studentCourse);

        //Assert
        Assert.AreEqual(0, results.Count);
```

Merima Durić 19025

Loop coverage:

Kada je lista prazna:

```
[TestMethod]
| 0 references | Please sign-in to New Relic CodeStream to see Code Level Metrics
public async Task Test3()
{
    // Arrange
    var student = new Student { Id = 1, FirstName = "Jane", LastName = "Doe", UserName = "janeDOe", Email = "jane@example.com", Index = 12345, Department = "RI", Year = 3 };
    var studentCourse = new List<StudentCourse>{};

    var controller = new StudentExamController(_mockContext.Object, _mockUserManager.Object);


    //Act
    var results = controller.GetFullNames(studentCourse);

    //Assert
    Assert.AreEqual(0, results.Count);
}
```

Kada lista sadrži više različitih elemenata:

```
[TestMethod]
| 0 references | Please sign-in to New Relic CodeStream to see Code Level Metrics
public async Task Test4()
{
    // Arrange
    var student = new Student { Id = 1, FirstName = "Jane", LastName = "Doe", UserName = "janeDOe", Email = "jane@example.com", Index = 12345, Department = "RI", Year = 3 };
    var studentCourse = new List<StudentCourse> {
    new StudentCourse { ID = 99, CourseID = 1, StudentID = 1,Student = student,  Points = 75, Grade = 8 },
    new StudentCourse { ID = 9, CourseID = 11, StudentID = 1,Student = student,  Points = 75, Grade = 8 },
    new StudentCourse { ID = 999, CourseID = 111, StudentID = 1,Student = student,  Points = 75, Grade = 8 },
    };

    var controller = new StudentExamController(_mockContext.Object, _mockUserManager.Object);


    //Act
    var results = controller.GetFullNames(studentCourse);

    //Assert
    Assert.AreEqual(3, results.Count);
}
```

Kada lista sadrži više istih elemenata:

```
[TestMethod]
| 0 references | Please sign-in to New Relic CodeStream to see Code Level Metrics
public async Task Test1()
{
    // Arrange
    var student = new Student { Id = 1, FirstName = "Jane", LastName = "Doe", UserName = "janeDOe", Email = "jane@example.com", Index = 12345, Department = "RI", Year = 3 };
    var studentCourse = new List<StudentCourse>

new StudentCourse { ID = 99, CourseID = 1, StudentID = 1,Student = student,  Points = 75, Grade = 8 },
new StudentCourse { ID = 99, CourseID = 1, StudentID = 1,Student = student,  Points = 75, Grade = 8 },
new StudentCourse { ID = 99, CourseID = 1, StudentID = 1,Student = student,  Points = 75, Grade = 8 },


    var controller = new StudentExamController(_mockContext.Object, _mockUserManager.Object);


    //Act
    var results = controller.GetFullNames(studentCourse);

    //Assert
    Assert.AreEqual(3, results.Count);
}
```

Kada lista sadrži jedan element:

```
    [TestMethod]
    | 0 references | Please sign-in to New Relic CodeStream to see Code Level Metrics
    public async Task Test2()
    {
        // Arrange
        var student = new Student { Id = 1, FirstName = "Jane", LastName = "Doe", UserName = "janeDOe", Email = "jane@example.com", Index = 12345, Department = "RI", Year = 3 };
        var studentCourse = new List<StudentCourse>
{
    new StudentCourse { ID = 99, CourseID = 1, StudentID = 1,  Points = 75, Grade = 8 }
};

        var controller = new StudentExamController(_mockContext.Object, _mockUserManager.Object);


        //Act
        var results = controller.GetFullNames(studentCourse);

        //Assert
        Assert.AreEqual(0, results.Count);
```

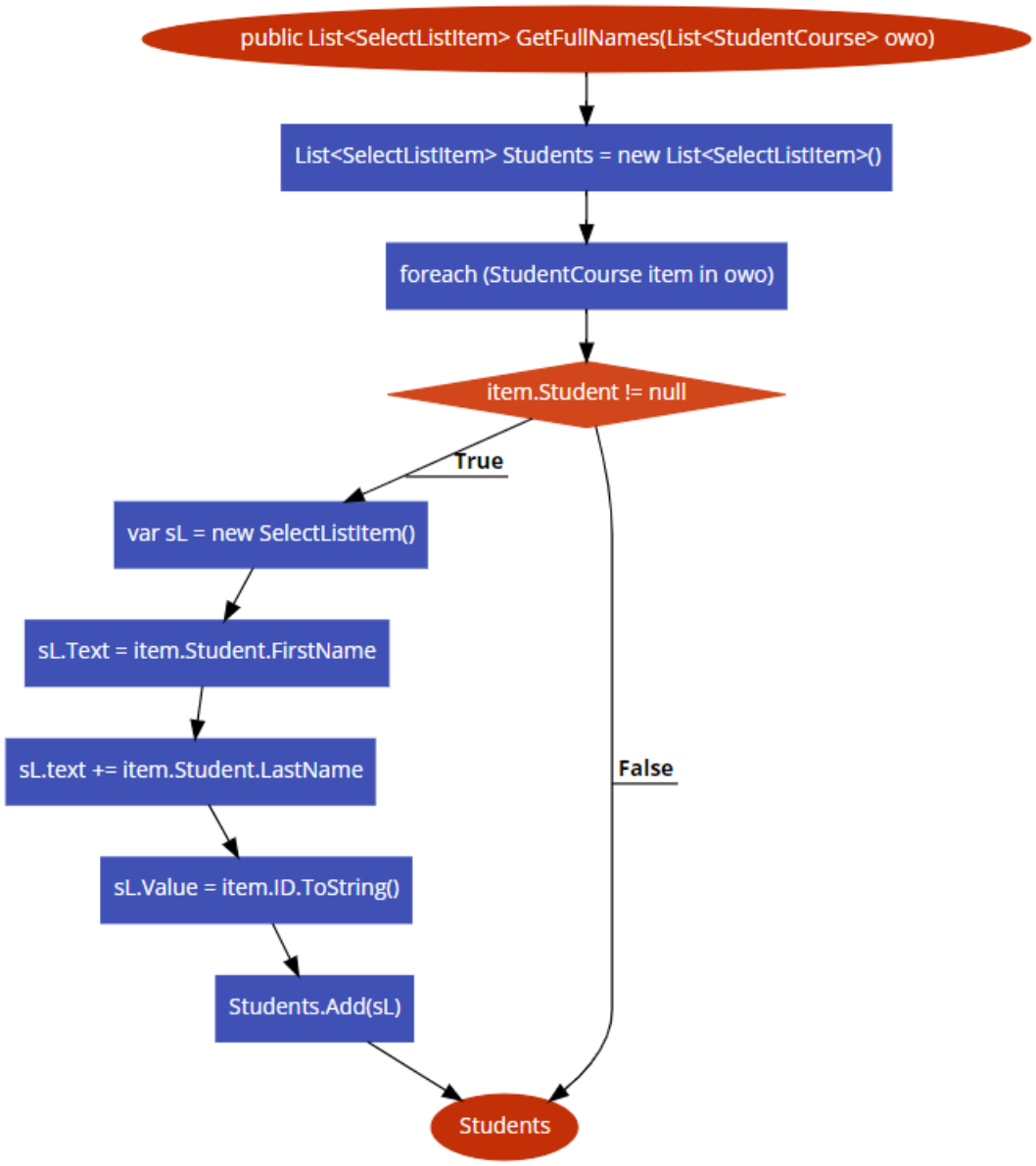
Merima Durić 19025

Ukoliko je jedan Student null:

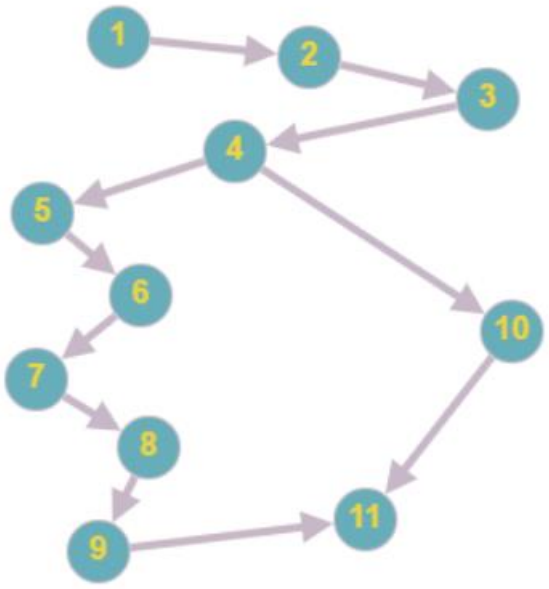
```
[TestMethod]
| 0 references | Please sign-in to New Relic CodeStream to see Code Level Metrics
public async Task Test5()
{
    // Arrange
    var student = new Student { Id = 1, FirstName = "Jane", LastName = "Doe", UserName = "janeDOe", Email = "jane@example.com", Index = 12345, Department = "RI", Year = 3 };
    Student student3 = null;
    var studentCourse = new List<StudentCourse> {
    new StudentCourse { ID = 99, CourseID = 1, StudentID = 1,Student = student3,  Points = 75, Grade = 8 },
    new StudentCourse { ID = 9, CourseID = 11, StudentID = 1,Student = student,  Points = 75, Grade = 8 },
    new StudentCourse { ID = 999, CourseID = 111, StudentID = 1,Student = student,  Points = 75, Grade = 8 },
    };

    var controller = new StudentExamController(_mockContext.Object, _mockUserManager.Object);


    //Act
    var results = controller.GetFullNames(studentCourse);

    //Assert
    Assert.AreEqual(2, results.Count);
}
```

Merima Durić 19025

Path coverage:



Merima Durić 19025



Obzirom da se metoda završava u čvoru 11, imamo dva moguća puta, što znači da su potrebna najmanje dva testa kako bi se postigla potpuna obuhvatnost puteva.

Put 1	1⇒2⇒3⇒4⇒10⇒11
Put 2	1⇒2⇒3⇒4⇒5⇒6⇒7⇒8⇒9⇒11

```
[TestMethod]
0 references | Please sign-in to New Relic CodeStream to see Code Level Metrics
public async Task TestPut1()
{
    // Arrange

    var studentCourse = new List<StudentCourse> {
    new StudentCourse { ID = 99, CourseID = 1, StudentID = 1,Student = null,  Points = 75, Grade = 8 },

    };

    var controller = new StudentExamController(_mockContext.Object, _mockUserManager.Object);


    //Act
    var results = controller.GetFullNames(studentCourse);

    //Assert
    Assert.AreEqual(0, results.Count);
}
```

Merima Durić 19025

```
[TestMethod]
0 references | Please sign-in to New Relic CodeStream to see Code Level Metrics
public async Task TestPut2()
{
    // Arrange
    var student = new Student { Id = 1, FirstName = "Jane", LastName = "Doe", UserName = "janeDOe", Email = "jane@example.com", Index = 12345, Department = "RI", Year = 3 };
    var studentCourse = new List<StudentCourse> {
    new StudentCourse { ID = 99, CourseID = 1, StudentID = 1,Student =student,  Points = 75, Grade = 8 },
    };

    var controller = new StudentExamController(_mockContext.Object, _mockUserManager.Object);


    //Act
    var results = controller.GetFullNames(studentCourse);

    //Assert
    Assert.AreEqual(1, results.Count);
}
```