Proje Raporu

Yazılım Sınama

XPTaskBoard

Hazırlayanlar

Ali Yıldızöz **172804001**

Hüseyin Balicak **152802027**

İçindekiler

Projede kullanılan teknoloji ve tasarım kalıpları **............. 3**

Proje büyüklüğü tahminlemesi **....................................... 4**

XPTaskBoard Proje metrikleri **......................................... 5**

Test caseler **.................................................................... 8**

Unit Test Raporu **........................................................... 11**

Github **........................................................................... 12**

Projede kullanılan teknoloji ve tasarım kalıpları.

* Windows İşletim sistemi
* Visual studio 2019
* C# dili
* MsSql veri tabanı.
* Entity Framework (v6.4.4).
* Windows Form UI
* Metro framework (v.1.4.0).
* Material Skin (v0.2.1).
* Fluent Validation(v9.3.0).
* Repository Patern
* Dependency Injection (AutoFac(v6.1.0)).
* N katmanlı mimari.
* NUnit

Proje büyüklüğü tahminlemesi

**Mantık**

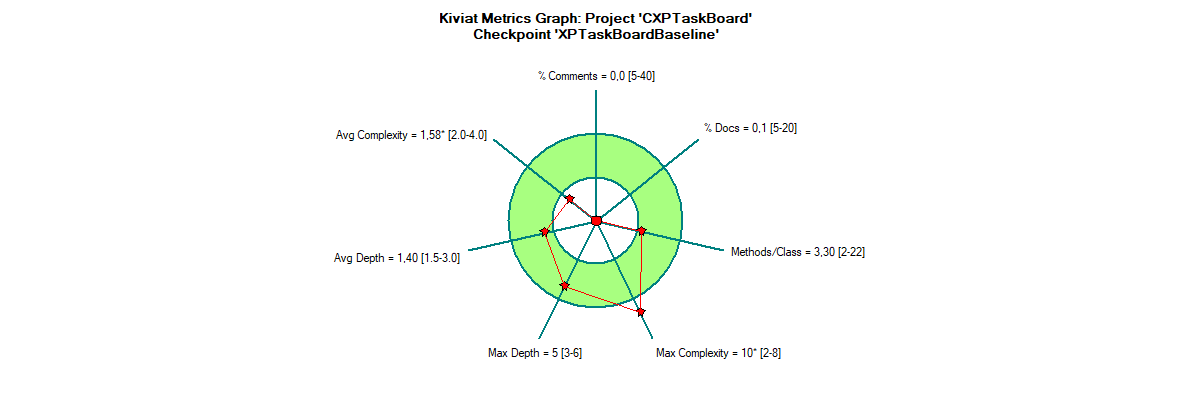
Projedeki taskların ortalama bitiş tarihi ile tahminleme yapılmıştır.

**Nasıl yapıldı (Algoritma) ?**

1. Projede ,şuanki durumu bitmiş olarak işaretlenmiş taskları bul.
2. Bu taskları bir **A** listesine doldur.
3. Listenin içini gez.
4. Her bir taskın bitiş tarihinden taskın oluşturulma tarihini çıkar.
5. Çıkan sonucu **B** listesine doldur.
6. **A** listesindeeleman bittiyse;
   1. **Evet : 7.Adıma git**
   2. **Hayır :** **4.Adıma git**
7. **B** listesindeki sayıların ortalamasını al **C** değişkenine ata.
8. **C** değişkenini üste yuvarla.
9. Bu günün tarihine **C** değişkeni kadar gün ekle ve bunu **return et**.

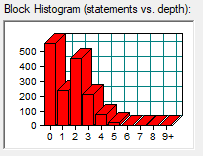
XPTaskBoard Proje metrikleri

**Kiviat metrics graph**

****

**Block Histogram**

|  |  |
| --- | --- |
| Block Depth | Statements |
| 0 | 552 |
| 1 | 235 |
| 2 | 449 |
| 3 | 210 |
| 4 | 74 |
| 5 | 19 |
| 6 | 0 |
| 7 | 0 |
| 8 | 0 |
| 9+ | 0 |



**Metrikler**

|  |  |
| --- | --- |
| Parameter | Value |
| Project Directory | C\Users\aliyi\source\repos\XPTaskBoard\ |
| Project Name | XPTaskBoard |
| Checkpoint Name | XPTaskBoardBaseline |
| Created On | 14 Jan 2021, 14 |
| 21 | 47 |
| Files | 59 |
| Lines | 2.329\* |
| Statements | 1.539 |
| Percent Comment Lines | 0,0 |
| Percent Documentation Lines | 0,1 |
| Classes, Interfaces, Structs | 57 |
| Methods per Class | 3,30 |
| Calls per Method | 2,68 |
| Statements per Method | 3,62 |
| Line Number of Most Complex Method | {undefined} |
| Name of Most Complex Method | TaskCard.OnChangeState(object sender, System.EventArgs e) |
| Maximum Complexity | 10\* |
| Line Number of Deepest Block | {undefined} |
| Maximum Block Depth | 5 |
| Average Block Depth | 1,40 |
| Average Complexity | 1,58\* |

|  |  |
| --- | --- |
| Most Complex Methods in 41 Class(es) | Complexity, Statements, Max Depth, Calls |
| AppHelper.GetStateColor(string desc) | 7\*, 13, 3, 6 |
| ChangeStateEventArgs.ChangeStateEventArgs(int taskId) | 1\*, 1, 2, 0 |
| CreateProject.BtnCreateProject\_Click(object sender, System.EventArgs e) | 4\*, 17, 4, 13 |
| DataResultBase<T>.DataResultBase(bool isSuccessful, T data) | 1\*, 2, 2, 0 |
| DI.GetService<T>() | 1\*, 4, 3, 2 |
| EditProject.BtnEditProject\_Click(object sender, System.EventArgs e) | 7\*, 18, 5, 9 |
| EfEntityRepositoryBase<TEntity, TContext>.Delete(TEntity entity) | 1\*, 4, 3, 3 |
| EfTaskDal.RemoveEmployee(int taskId, Employee employee) | 1\*, 2, 3, 2 |
| Employee.ToString() | 1\*, 1, 2, 0 |
| EmployeeCreate.BtnCreateEmployee\_Click(object sender, System.EventArgs e) | 4\*, 8, 3, 7 |
| EmployeeEdit.BtnUpdateEmployee\_Click(object sender, System.EventArgs e) | 8\*, 12, 4, 6 |
| EmployeeManager.Search(string search) | 2\*, 1, 2, 10 |
| EmployeeValidator.EmployeeValidator() | 1\*, 2, 2, 6 |
| EntityBase.EntityBase() | 1\*, 1, 2, 0 |
| ErrorDataResult<T>.ErrorDataResult() | 1\*, 0, 0, 0 |
| ErrorResult.ErrorResult() | 1\*, 0, 0, 0 |
| MainForm.LoadData() | 7\*, 26, 4, 10 |
| Program.Main() | 1\*, 3, 2, 3 |
| Project.ToString() | 1\*, 1, 2, 0 |
| ProjectManager.Search(string search) | 1\*, 1, 2, 7 |
| ProjectValidator.ProjectValidator() | 1\*, 1, 2, 0 |
| RegisterIoC.Load(ContainerBuilder builder) | 1\*, 9, 2, 18 |
| ResultBase.Message.get() | 1\*, 0, 0, 0 |
| SelectProject.BtnSelect\_Click(object sender, System.EventArgs e) | 4\*, 14, 4, 9 |
| Session.SetCurrentProject(Project project) | 1\*, 1, 2, 0 |
| SuccessDataResult<T>.SuccessDataResult() | 1\*, 0, 0, 0 |
| SuccessResult.SuccessResult() | 1\*, 0, 0, 0 |
| Task.ToString() | 1\*, 1, 2, 0 |
| TaskCard.OnChangeState(object sender, System.EventArgs e) | 10\*, 16, 4, 1 |
| TaskCreate.OnChangeState(object sender, System.EventArgs e) | 7\*, 7, 3, 0 |
| TaskDetails.btnDeleteTask\_Click(object sender, System.EventArgs e) | 4\*, 8, 4, 5 |
| TaskEdit.OnCheckedChanged(object sender, System.EventArgs e) | 7\*, 10, 3, 0 |
| TaskManager.GetByIdWithTaskTracks(int id) | 2\*, 4, 3, 3 |
| TaskState.ToString() | 1\*, 1, 2, 0 |
| TaskStateValidator.TaskStateValidator() | 1\*, 1, 2, 0 |
| TaskTrack.Task.set() | 1\*, 0, 0, 0 |
| TaskValidator.TaskValidator() | 1\*, 1, 2, 0 |
| Validation<TValidator>.Validate(object entity) | 2\*, 6, 3, 3 |
| ValidatorFactory.CreateInstance(Type validatorType) | 1\*, 1, 2, 0 |
| XPTaskBoardDbContext.TaskTracks.set() | 1\*, 0, 0, 0 |
| XPTaskBoardDBInitializer.Seed(XPTaskBoardDbContext context) | 1\*, 13, 2, 20 |

|  |  |
| --- | --- |
| **Risk Level** | Yüksek |
| **Purpose** | Yeni bir taskın yaratılmasını denetlemek. |
| **Inputs** | 1. Details **RichTextBoxtan** değer girme. 2. Notes **RichTextBoxtan** değer girme. 3. State **RadioButtondan** state seçme. 4. **Create buttonuna** tıklama**.** |
| **Expected Outputs** | Taskın veri tabanına kaydedilmesi sonucunda IsSuccessFul değerinin **true** gelmesi ve panelde taskın görüntülenmesi. |
| **Pass Criterias** | State değerine göre panelde yaratılan taskın görüntülenmesi. |
| **Fail Criterias** | 1. Veri tabanına kaydın yapılamaması. 2. Panelde gösterilememesi. |
| **Test Procedure** | 1. Uygulama başlatıldı. 2. Bir proje seçildi. 3. **Create new task** butonuna tıklandı. 4. Task oluşturma penceresi açıldı. 5. Details,Notes ve state değerleri girildi. 6. **Create** butonuna tıklandı. 7. Panelde ve veri tabanında oluşturulan task görüntülendi. |
| **Durum** | **Test Başarılı.** |

Test caseler

Test-case 1

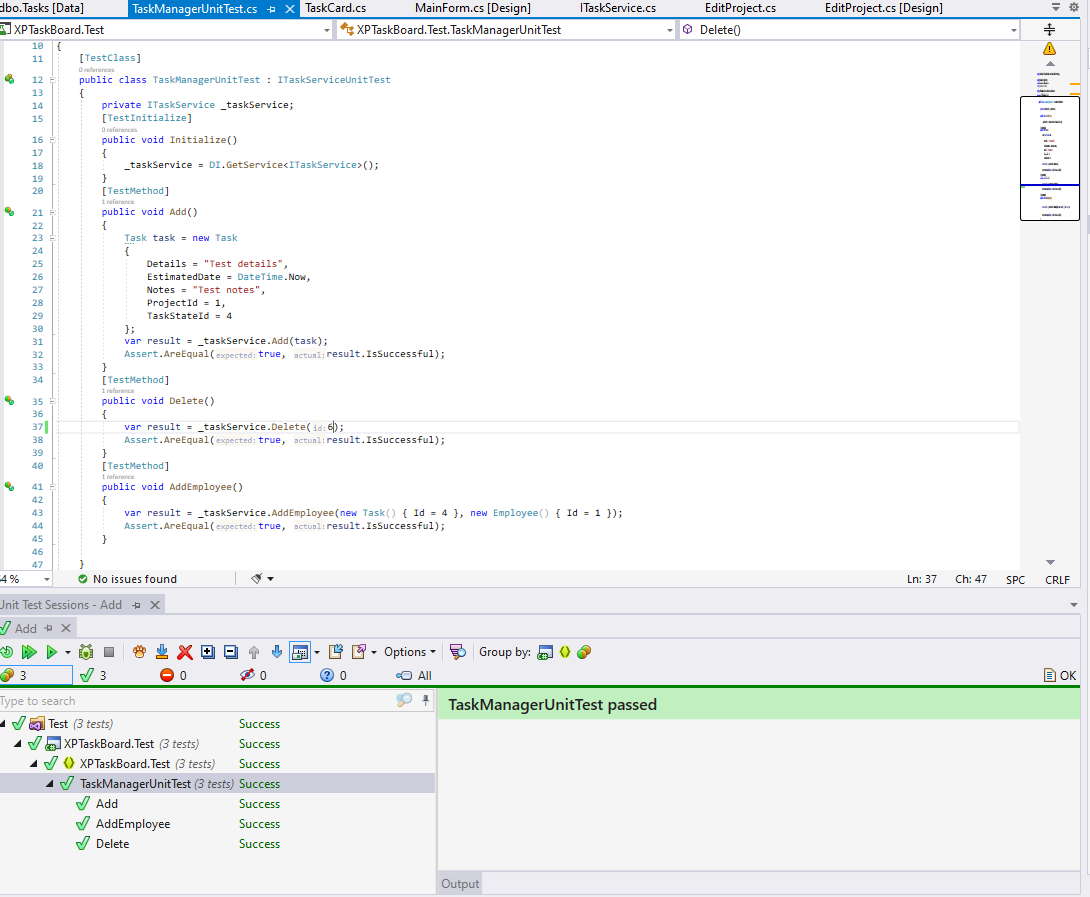
Test-case 2

|  |  |
| --- | --- |
| **Risk Level** | Yüksek |
| **Purpose** | Mevcut bir taska bir çalışan atama işlemini denetleme. |
| **Inputs** | 1. Employees **ComboBox tan** çalışan seçme. 2. **“+” ikonlu yeşil butonuna** tıklama. |
| **Expected Outputs** | Çalışanın ilgili taska eklenmesi sonucunda IsSuccessFul değerinin **true** gelmesi ve **ListBox** Controlunde görüntülenmesi. |
| **Pass Criterias** | **ListBox** Controlünde eklenen taskın görüntülenmesi ve veri tabanına kaydın yapılması. |
| **Fail Criterias** | 1. Veri tabanına kaydın yapılamaması. 2. **ListBoxta** kaydın görüntülenememesi. |
| **Test Procedure** | 1. Uygulama başlatıldı. 2. Bir proje seçildi. 3. En yakın taskın üzerindeki **details** butonuna tıklandı. 4. Task details ekranı açıldı. 5. **Combobox** controlüne tıklandı. 6. Bir çalışan seçildi. 7. **“+” ikonlu yeşil** butona tıklandı. 8. ListBox Controlünde seçilen çalışanın görüntülendi. |
| **Durum** | Test Başarılı. |

|  |  |
| --- | --- |
| **Risk Level** | Yüksek |
| **Purpose** | Bir taskın silinmesini denetlemek. |
| **Inputs** | **Çöp kutusu ikonlu kırmızı butonuna** tıklama**.** |
| **Expected Outputs** | Taskın veri tabanından silinmesi sonucunda IsSuccessFul değerinin **true** gelmesi ve panelde taskın görüntülenmemesi. |
| **Pass Criterias** | Panelde yaratılan taskın görüntülenmemesi. |
| **Fail Criterias** | 1. Veri tabanına taskın görüntülenmesi. 2. Panelde gösterilmesi. |
| **Test Procedure** | 1. Uygulama başlatıldı. 2. Bir proje seçildi. 3. En yakın taskın üzerindeki **details** butonuna tıklandı. 4. Task details ekranı açıldı. 5. **Çöp kutusu ikonlu kırmızı** **butonuna** tıklandı. 6. Panelde silinen ve veri tabanında task görüntülenmedi. |
| **Durum** | Test Başarılı. |

Test-case 3

Unit Test Raporu



Github

[github.com/AliYildizoz909/XPTaskBoard](https://github.com/AliYildizoz909/XPTaskBoard)

**Not : Veritabanı normalizasyonu ve yazılım şartnamesi aynı klasördedir.**