

Simple Jira Project

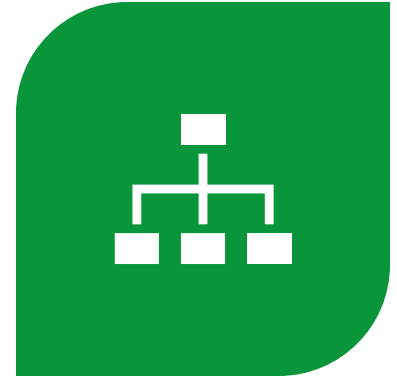
Team 9: Ying LUO, Khalil Hanna

[Detailed Presentation Source in Confluence](#)

Background



TASK MANAGEMENT SOFTWARE IS ANY DIGITAL TOOL THAT HELPS YOU PLAN, TRACK, AND COMPLETE A UNIT OF WORK



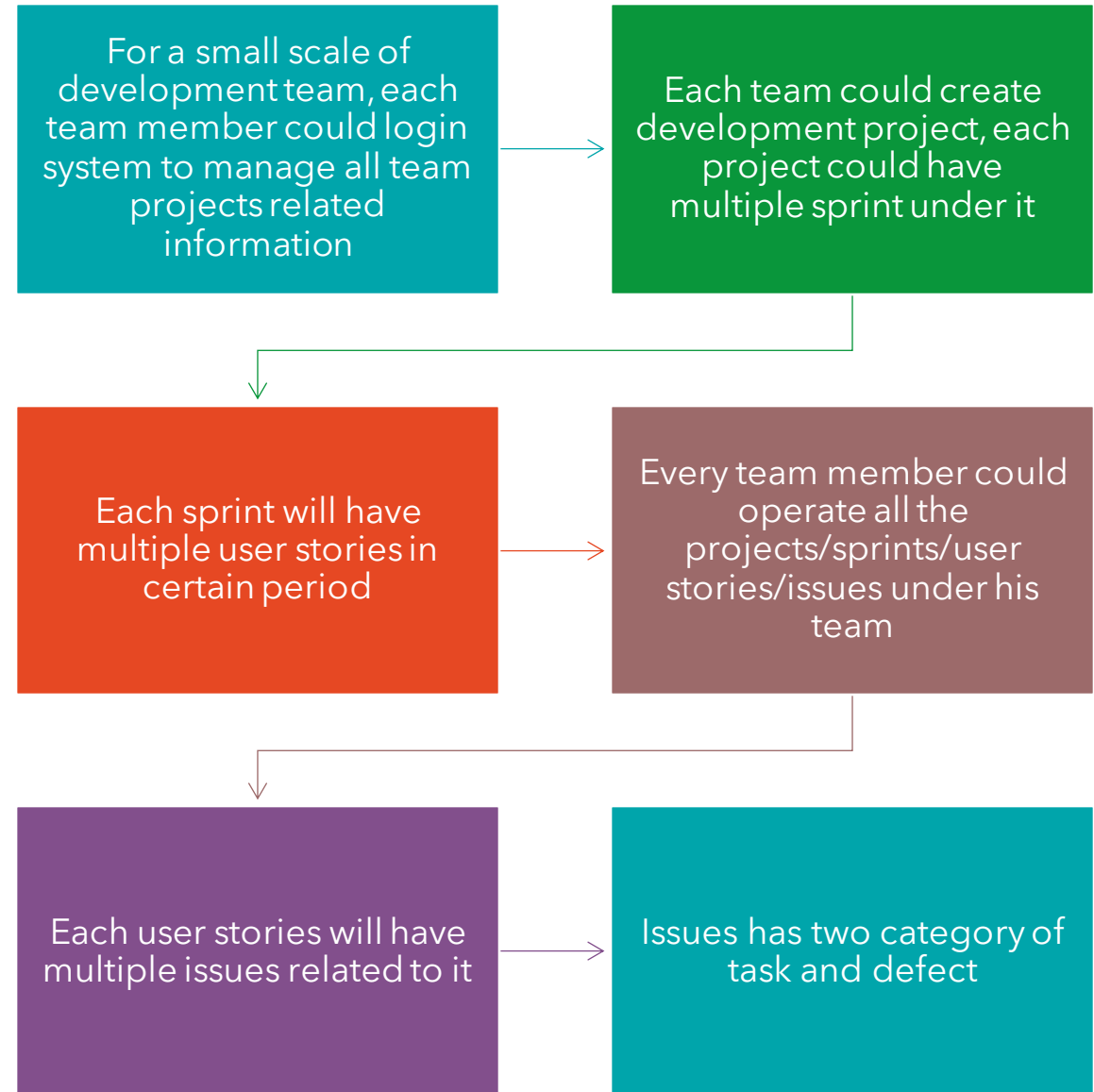
OFTEN HELP TO LIST, EXPENSE, AND ORGANIZE A GIVEN TASK OR SUB-TASK.



High Level View

- Simple task management, including project, sprint, user story and issues, sprint could be broke down by many user stories, and user stories can be related to many issues
- Team assignment for each project, Owner assigned to sprint / user story / issue (task/defect)
- CRUD operations for all entities

Business Logic



Resource Management



Project Management:

[Trello Board -- Tracking issue cards](#)



Source Control:

[Bitbucket](#)



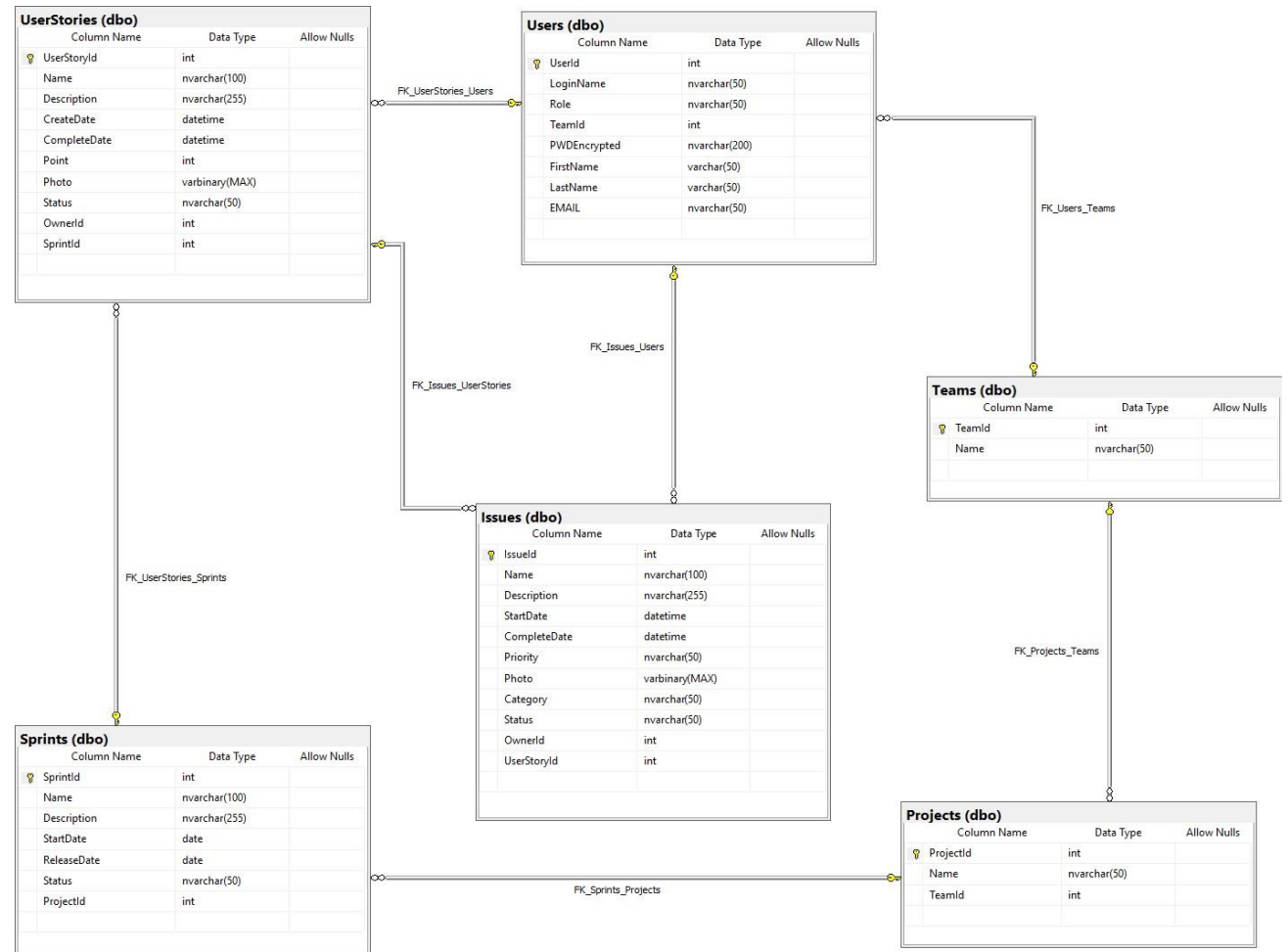
Knowledge Management:

[Confluence](#)

Main windows

| Simple Jira Application | | | | | | | |
|-------------------------|--|-------------|---------------|-------|-------------|--------|-------------|
| Name | Description | Create Date | Complete Date | Point | Sprint | Owner | Status |
| New requireme | As a tester of this application, I need a system design for review | 12/21/2020 | 1/20/2021 | 80 | Sprint 2020 | David | TEST |
| Another user st | wdvwdvd | 1/16/2021 | 1/31/2021 | 20 | Sprint 2020 | Defect | Documenting |
| User story for t | ghjgj hjgj jhg ghj | 1/14/2021 | 1/30/2021 | 55 | Sprint 2020 | Defect | DONE |
| gigjg | ghjgjgj | 1/14/2021 | 1/22/2021 | 22 | Sprint-1 | Defect | DONE |

Database Structure



Libraries



MaterialDesignThemes by James Willock, **1.68M** downloads

v3.2.0

ResourceDictionary instances containing Material Design templates and styles for WPF controls in the MahApps library.



MaterialDesignColors by James Willock, **1.27M** downloads

v1.2.7

ResourceDictionary instances containing standard Google Material Design swatches, for inclusion in a XAML application.



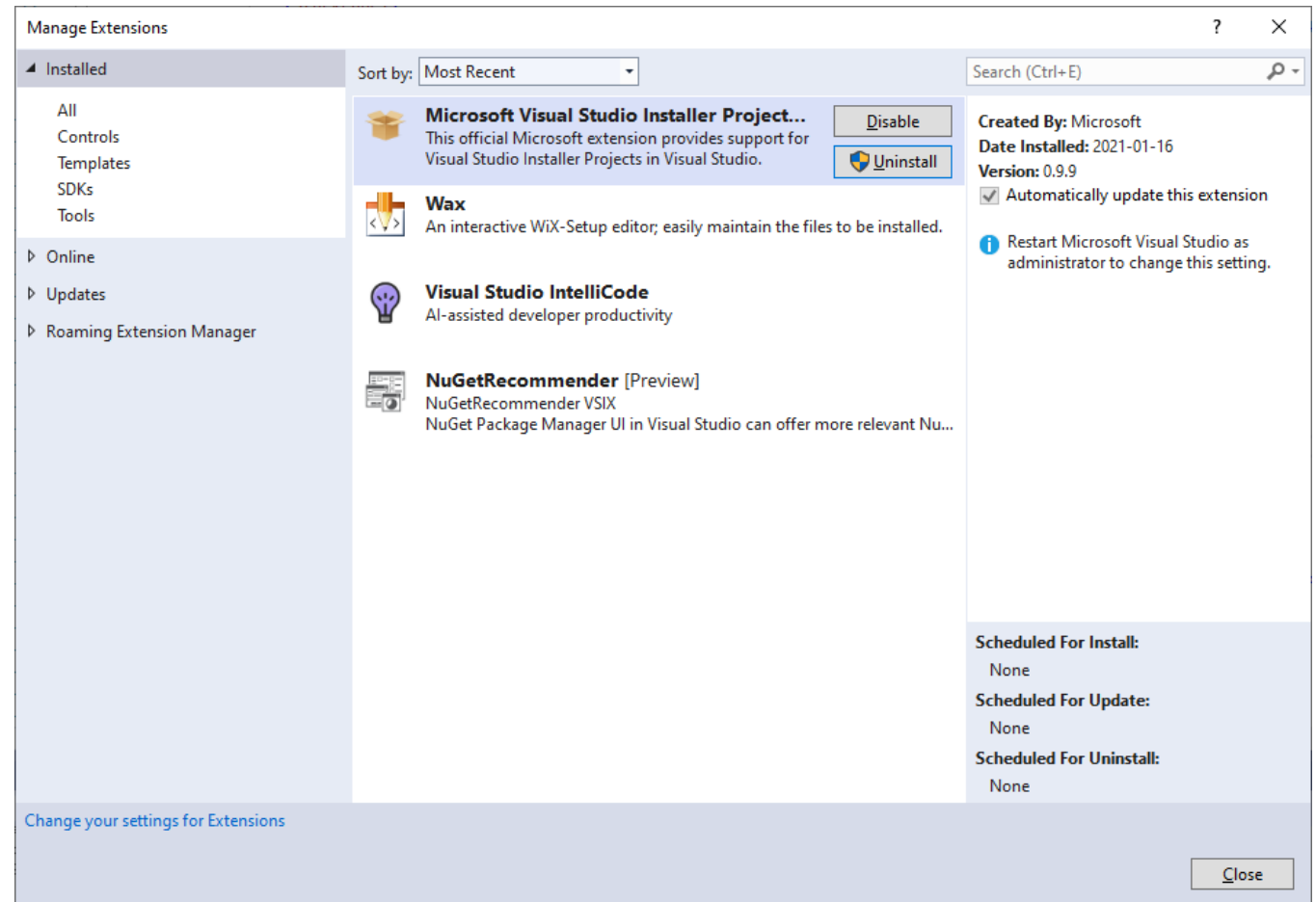
Dragablz by James Willock, **453K** downloads

v0.0.3.223

Dragable and tearable tab control for WPF

[Material Design In XAML](#) and [Dragablz for TabControl](#)

MS VS Installer Project



MS Unit Test

The screenshot displays the Test Explorer interface in Visual Studio. The top toolbar shows 15 tests passed (green checkmarks), 0 tests failed (red X), and 0 tests were skipped (grey X). The test tree on the left is expanded, showing the following structure:

- SimpleJiraProject.UnitTests (15) [23 ms]
 - SimpleJiraProject.UnitTests (15) [23 ms]
 - UnitTest (15) [23 ms]
 - Validate_RightDate_ReturnTrue [< 1 ms]
 - Validate_RightDescription_ReturnTrue [< 1 ms]
 - Validate_RightEmail_ReturnTrue [4 ms]
 - Validate_RightName_ReturnTrue [< 1 ms]
 - Validate_RightPassword_ReturnTrue [< 1 ms]
 - Validate_RightPoint_ReturnTrue [< 1 ms]
 - Validate_WrongDate_ReturnFalse [< 1 ms]
 - Validate_WrongDescription_ReturnFalse [2 ms]
 - Validate_WrongEmail_ReturnFalse [2 ms]
 - Validate_WrongLongName_ReturnFalse [< 1 ms]
 - Validate_WrongLongName_ThrowExcep... [9 ms]
 - Validate_WrongName_ReturnFalse [< 1 ms]
 - Validate_WrongName_ThrowException [< 1 ms]
 - Validate_WrongPassword_ReturnFalse [6 ms]
 - Validate_WrongPoint_ReturnFalse [< 1 ms]

The right-hand pane shows the 'Group Summary' for 'SimpleJiraProject.UnitTests':

- Tests in group: 15
- Total Duration: 23 ms
- Outcomes: 15 Passed

Challenges and Solutions

- User Login / Logout

▼ App.xaml

```
1 <Application x:Class="SimpleJiraProject.App"
2             xmlns:local="clr-namespace:SimpleJiraProject"
3             ShutdownMode="OnExplicitShutdown"
4             StartupUri="LoginDialog.xaml" >
```

Challenges and Solutions

- Secure Password by Advanced Encryption Standard (AES) Algorithm

▼ Encrypt/Decrypt

C# – Encrypt/Decrypt Password

```
1 // Encrypt plain text
2 public static string Encrypt(string strData)
3 {
4     return Convert.ToBase64String(Encrypt(Encoding.UTF8.GetBytes(strData)));
5 }
6
7 // Create AesManaged, Encryptor, MemoryStream, CryptoStream from MemoryStream and Encrypter and write it.
8 public static byte[] Encrypt(byte[] strData)
9 {
10     // Generate password, which will be used to derive the key.
11     PasswordDeriveBytes passbytes = new PasswordDeriveBytes(strPermutation, new byte[] { bytePermutation1,
12                                                                                             bytePermutation2,
13                                                                                             bytePermutation3,
14                                                                                             bytePermutation4});
15
16     // Create Aes that generates a new key and initialization vector (IV).
17     MemoryStream memstream = new MemoryStream();
18     Aes aes = new AesManaged();
19     aes.Key = passbytes.GetBytes(aes.KeySize / 8);
20     aes.IV = passbytes.GetBytes(aes.BlockSize / 8);
21
22     CryptoStream cryptostream = new CryptoStream(memstream, aes.CreateEncryptor(), CryptoStreamMode.Write);
23     cryptostream.Write(strData, 0, strData.Length);
24     cryptostream.Close();
25     return memstream.ToArray();
26 }
27
28 //Call Encrypt
29 PWDEncrypted = SecurePassword.Encrypt
    (tbConfirmPassword.Password);
```

Challenges and Solutions

- User Management

▼ Add team automatically if signup user input a new team

```
1 public int Team_Check(string team)
2     {
3         Team chooseTeam = Globals.simpleJiraDB.Teams.Where(t => t.Name.Equals(team)).FirstOrDefault();
4         int teamId = chooseTeam != null ? chooseTeam.TeamId : 0;
5         if (teamId == 0)
6         {
7             Team newTeam = new Team { Name = team };
8             Globals.simpleJiraDB.Teams.Add(newTeam);
9             Globals.simpleJiraDB.SaveChanges();
10            return newTeam.TeamId;
11        }
12        else
13        {
14            return chooseTeam.TeamId;
15        }
16    }
```

Challenges and Solutions

- Different view according to different team

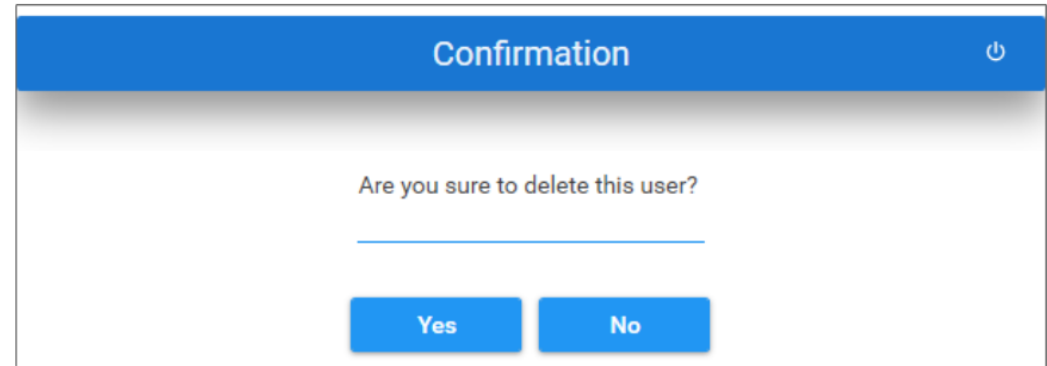
▼ Filter data by LINQ from database

```
1  Globals.currentTeamProjectList = Globals.simpleJiraDB.Projects.Where(p => p.TeamId == currentUser.TeamId).ToList<Project>();
2  ProjectListView.ItemsSource = Globals.currentTeamProjectList;
3
4  Globals.currentTeamUserList = Globals.simpleJiraDB.Users.Where(u => u.TeamId == currentUser.TeamId).ToList<User>();
5
6  IEnumerable<int> projectIds = Globals.currentTeamProjectList.Select(p => p.ProjectId).Distinct();
7
8  Globals.currentSprintList = Globals.simpleJiraDB.Sprints.Where(s => projectIds.Contains(s.ProjectId)).ToList<Sprint>();
9  IEnumerable<int> sprintIds = Globals.currentSprintList.Select(sp => sp.SprintId).Distinct();
10
11 SprintListView.ItemsSource = Globals.currentSprintList;
12
13 Globals.currentUserStoryList = Globals.simpleJiraDB.UserStories.Where(us => sprintIds.Contains(us.SprintId)).ToList<UserStory>();
14 IEnumerable<int> userStoryIds = Globals.currentUserStoryList.Select(us => us.UserStoryId).Distinct();
15
16 UserStoryListView.ItemsSource = Globals.currentUserStoryList;
17
18 Globals.currentIssueList = Globals.simpleJiraDB.Issues.Where(i => userStoryIds.Contains(i.UserStoryId)).ToList<Issue>();
```


Challenges and Solutions

- Custom MessageBox

▼ Custom MessageBox Dialog and usage



```
1 new MessageBoxCustom("Are you sure to delete this user?", MessageBoxCustom.MessageType.Warning, MessageBoxCustom.MessageButtons.Ok).ShowDialog();
```

Challenges and Solutions

- Performance Issue

⚠ Program getting slow when loading data from database because of large image saved into table

🔍 By operation to identify the problematic part of code, review the code to confirm the slow down caused by image loading from database

✅ Resolve by

- Late and separated loading image from database till the final operation
- Replace the object list view by exclude image list
- Image size constraint

```
1      IssueListView.ItemsSource = Globals.simpleJiraDB.Issues.Select(x => new IssueListItem
2      {
3          IssueId = x.IssueId,
4          Name = x.Name,
5          Description = x.Description,
6          StartDate = x.StartDate,
7          CompleteDate = x.CompleteDate,
8          Priority = x.Priority,
9          Status = x.Status,
10         Category = x.Category,
11         OwnerId = x.OwnerId,
12         UserStoryId = x.UserStoryId,
13         User = x.User,
14         UserStory = x.UserStory
15     }).Where(iss => userStoryIds.Contains(iss.UserStoryId)).ToList();
16     static long threshold = 2621440;    // 2.5MB
17     var fileLength = new FileInfo(fileName).Length;
18
19     if(fileLength < threshold)
20     {
21         if (currentIssue != null)
22         {
23             currentIssue.Photo = File.ReadAllBytes(fileName);
24             image.Source = new BitmapImage(new Uri(fileName));
25         }
26     }
```

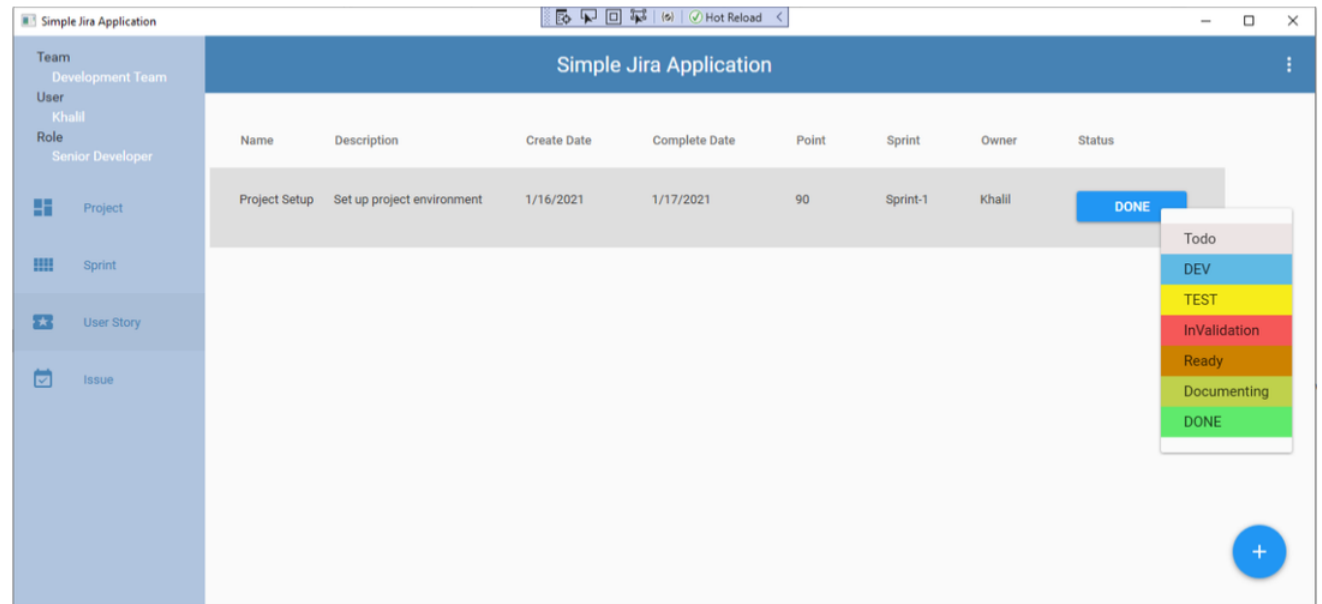
Challenges and Solutions

- Status Update - color set and save change to database quickly

▼ Purpose of implementation to have quick access to object status for frequent change

Challenges:

- Add item to list view
- Control the status



Challenges and Solutions

- User Control

Q Search

```
2      </GridViewColumn.CellTemplate>
3      <DataTemplate>
4          <local:UserStoryStatus />
5      </DataTemplate>
6  </GridViewColumn.CellTemplate>
7  </GridViewColumn>
```

```
1  <UserControl x:Class="SimpleJiraProject.UserStoryStatus"
2      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
5      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
6      xmlns:local="clr-namespace:SimpleJiraProject"
7      mc:Ignorable="d"
8      d:DesignHeight="450" d:DesignWidth="800">
9      <Grid>
10         <Button Name="btStatus" HorizontalContentAlignment="Center" ToolTip="Right Click to update the Status" MinWidth="115" >
11             <TextBlock Name="tbStatus" Text="{Binding Status, Mode=OneWay}" Height="18" HorizontalAlignment="Center" />
12         </Button>
13         <Grid.ContextMenu>
14             <ContextMenu>
15                 <MenuItem Header="Todo" Name="miTodo" Background="#FFCE4E4" Click="miTodo_Click"/>
16                 <MenuItem Header="DEV" Name="miDEV" Background="#FF60BAE4" Click="miDEV_Click"/>
17                 <MenuItem Header="TEST" Name="miTEST" Background="#FFF7ED1B" Click="miTEST_Click"/>
18                 <MenuItem Header="InValidation" Name="miInValidation" Background="#FFF55858" Click="miInValidation_Click"/>
19                 <MenuItem Header="Ready" Name="miReady" Background="#FFEA9500" Click="miReady_Click"/>
20                 <MenuItem Header="Documenting" Name="miDocumenting" Background="#FFBFD14C" Click="miDocumenting_Click"/>
21                 <MenuItem Header="DONE" Name="miDONE" Background="#FF5FEA6C" Click="miDONE_Click"/>
22             </ContextMenu>
23         </Grid.ContextMenu>
24     </Grid>
25 </UserControl>
```

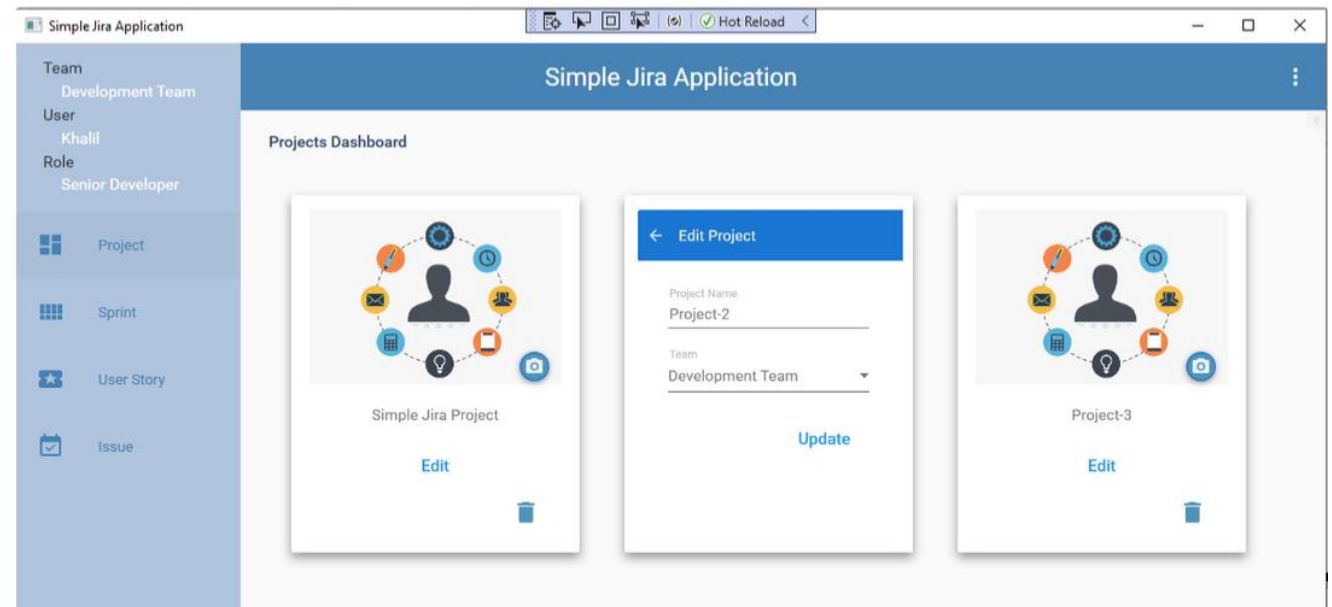
Challenges and Solutions

- Project view - implemented by card rather than by list view

▼ Purpose of implementation to have better UX than list view to manage the Projects

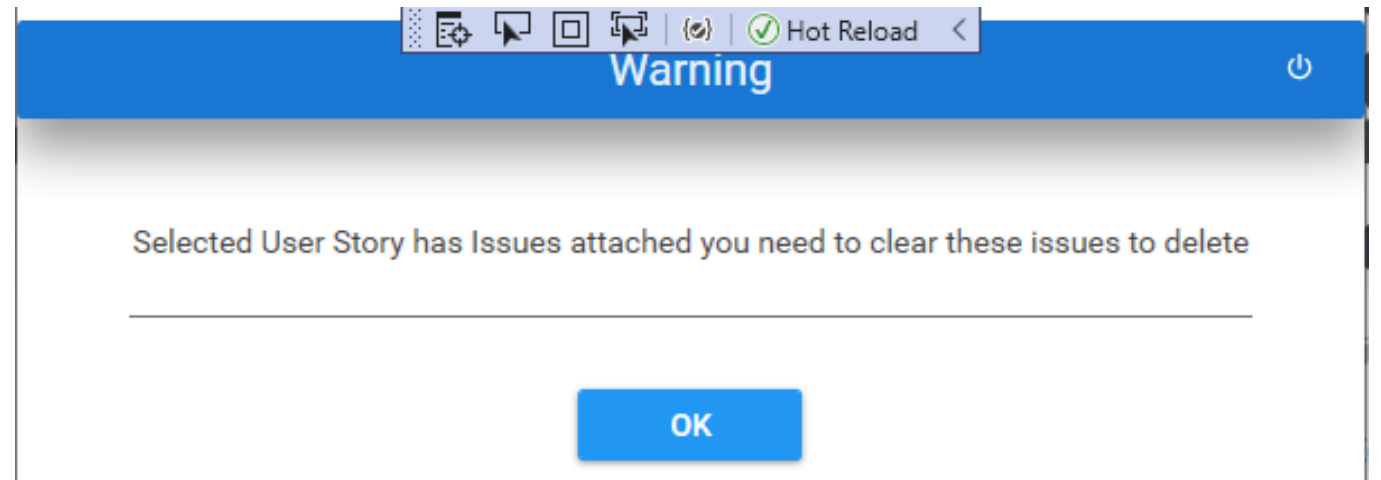
Challenges:

- Card Design
- Cards Creation
- Card Control



Challenges and Solutions

- Delete Object - Handle delete related objects



Future Work



Provide search function in each list of view and main window



Statistic chart generated for each list of view



Add Project Themes



Enhance the user Roles to optimize the Teams works



UI/UX enhancement

Summary

Designing 6 tables database with 1-Many relationships, practiced **Microsoft SQL server**

Using **C#** entity framework to create **WPF** project by database-first approach

Practicing **C# OOP** skills and event handling

Practicing C#, WPF XAML

Adapt **Material Design** UI library on WPF

Create installer by MS VS installer project

MS Unit Test for API testing