

# Simple Jira Project Presentation

## Title

## Simple Jira Project

## Subject

To provide a simple version of Jira which is a project task management tool for Agile software development.

## Team Member

@ Ying LUO

@ Khalil Hanna

## Background

## High level View

**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.

## Jira Issue Management

- 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

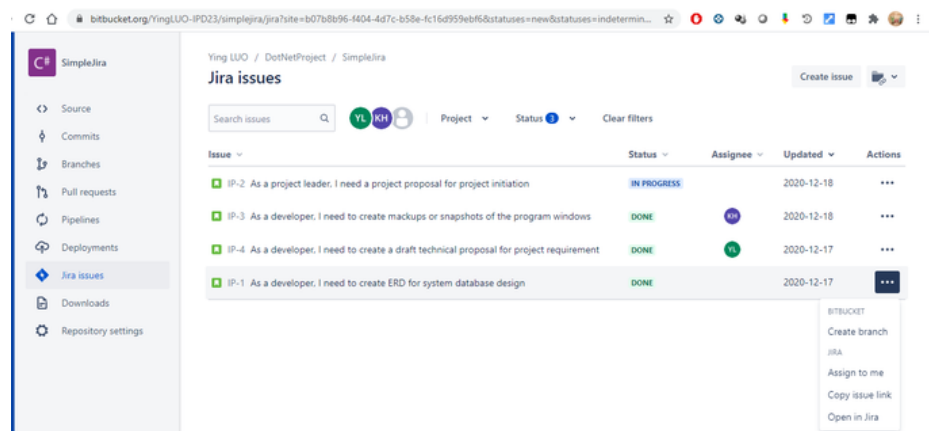
Idea comes from the Jira Issues integrated with bitbucket

Jira - The #1 software development tool used by agile teams

Jira Issues List Example

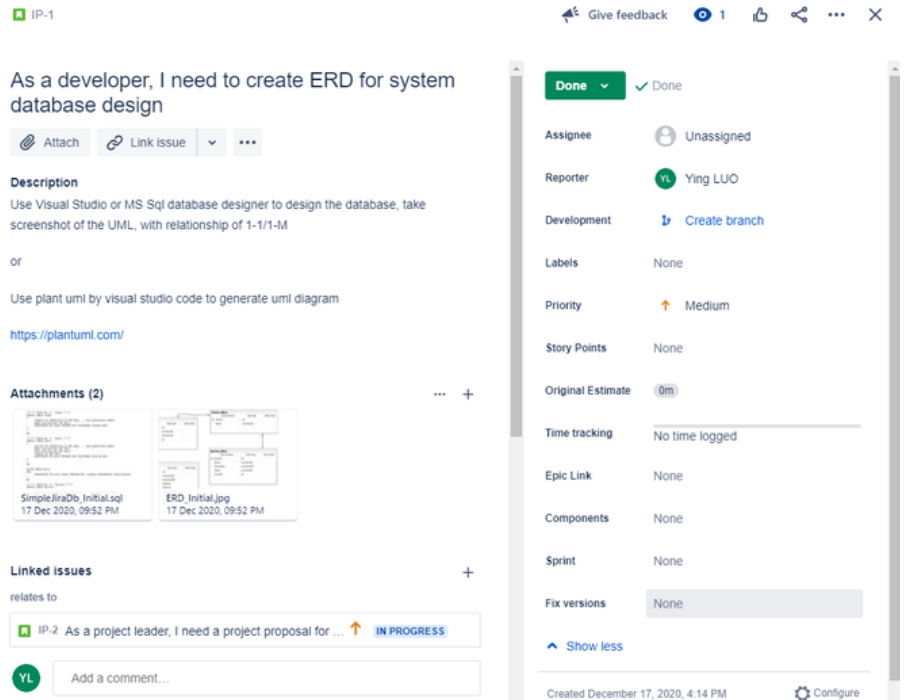
Main page of issue list

▼ [Main page](#)



By click each Issue, go to the detail page

▼ [Detailed page](#)



## Project Management:

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

## Source Control:







[Bitbucket](#)

## Knowledge Management:

[Confluence](#)

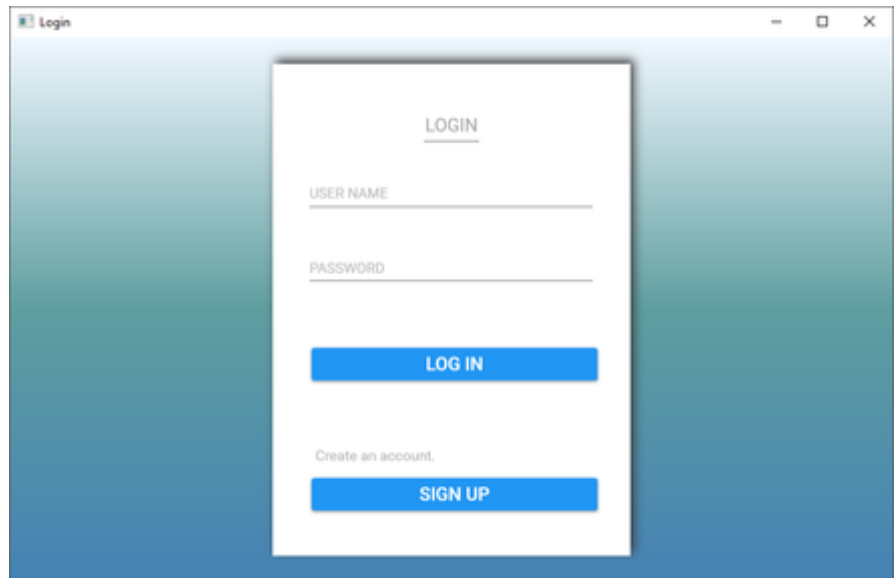
## Overview Solution

### Business Logic

-  For a small scale of development team, each team member could login system to manage all team projects related information
-  Each team could create development project, each project could have multiple sprint under it
-  Each sprint will have multiple user stories in certain period
-  Every team member could operate all the projects/sprints/user stories/issues under his team
-  Each user stories will have multiple issues related to it
-  Issues has two category of task and defect

### Main window/dialog Screenshots

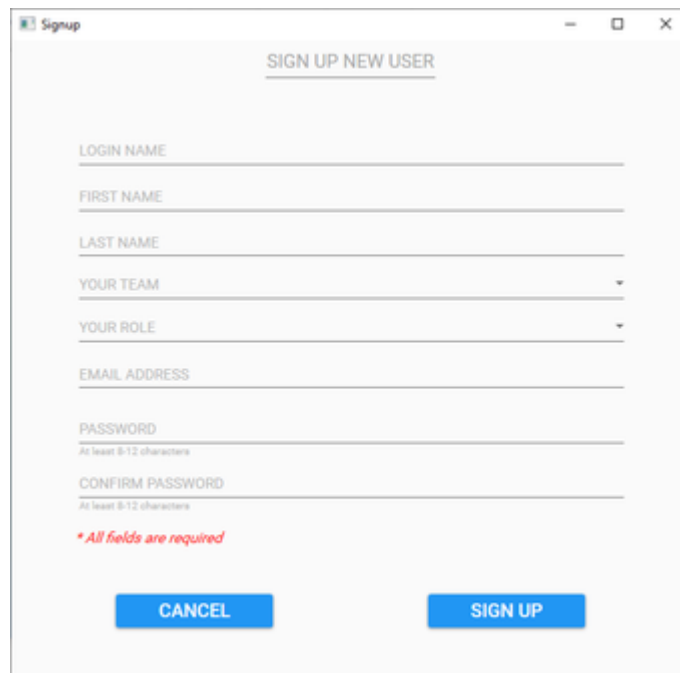
- Login with user name and password, or create new user by signup
- ▼ [Traditional User Login](#)



A screenshot of a web application window titled "Login". The window has a blue gradient background. In the center is a white rectangular form. At the top of the form is the word "LOGIN" in blue, underlined. Below it are two input fields: "USER NAME" and "PASSWORD", both with blue placeholder text. Under the "PASSWORD" field is a blue button with the text "LOG IN" in white. Below the button is the text "Create an account." in a smaller font. At the bottom of the form is another blue button with the text "SIGN UP" in white.

- Automatically create new Team and new role if they are not in the list from editable combo box

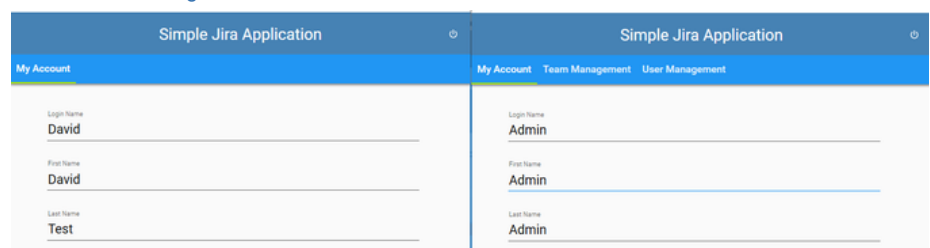
▼ SignUp



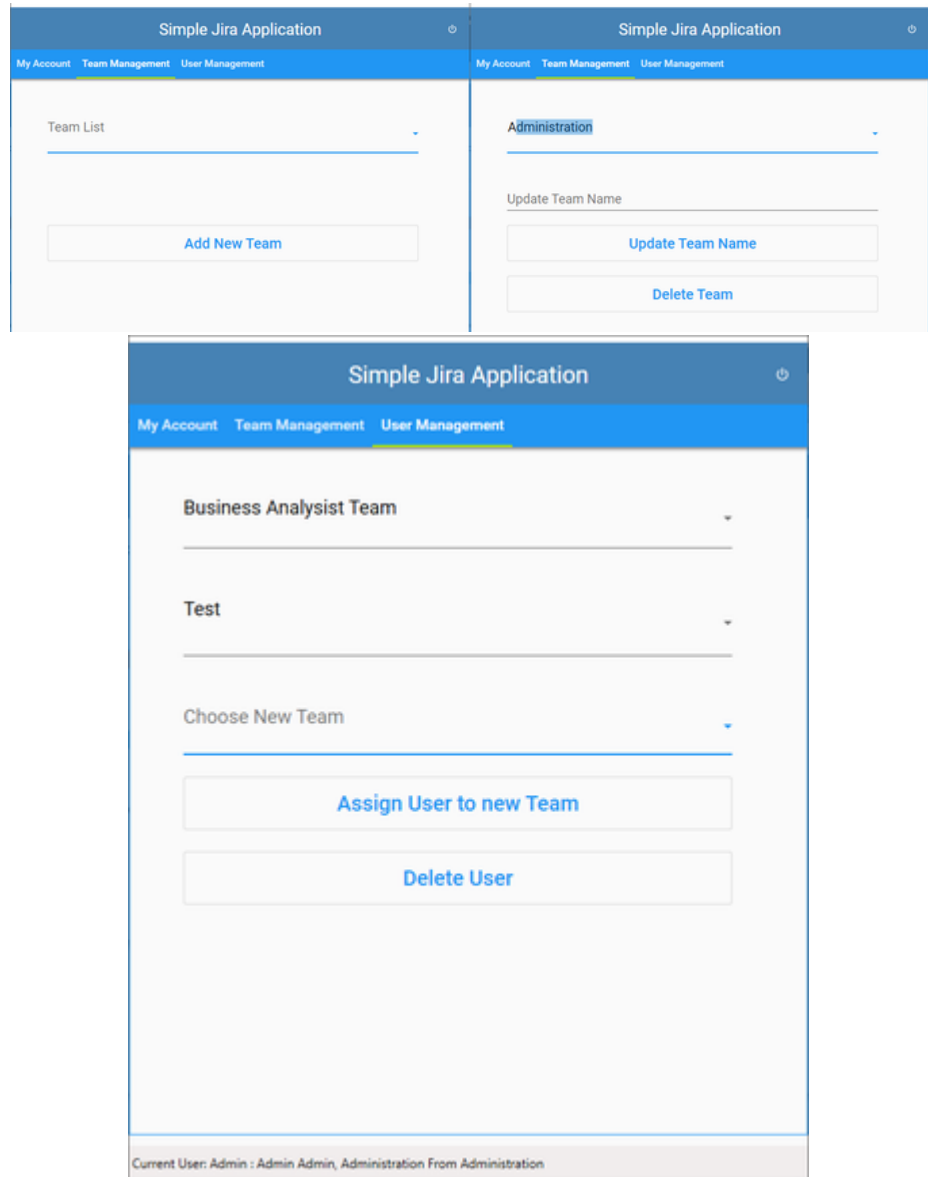
A screenshot of a web application window titled "Signup". The window has a light gray background. In the center is a white rectangular form. At the top of the form is the text "SIGN UP NEW USER" in blue, underlined. Below it are several input fields: "LOGIN NAME", "FIRST NAME", "LAST NAME", "YOUR TEAM" (with a dropdown arrow), "YOUR ROLE" (with a dropdown arrow), "EMAIL ADDRESS", "PASSWORD" (with a small text "At least 8-12 characters" below it), and "CONFIRM PASSWORD" (with a small text "At least 8-12 characters" below it). Below the input fields is a red line of text: "\* All fields are required". At the bottom of the form are two blue buttons: "CANCEL" on the left and "SIGN UP" on the right.

- Team/User/Account Management:
  - Different View by different login user
  - Add/update/delete team by different input of editable combo box

▼ Team & User Management

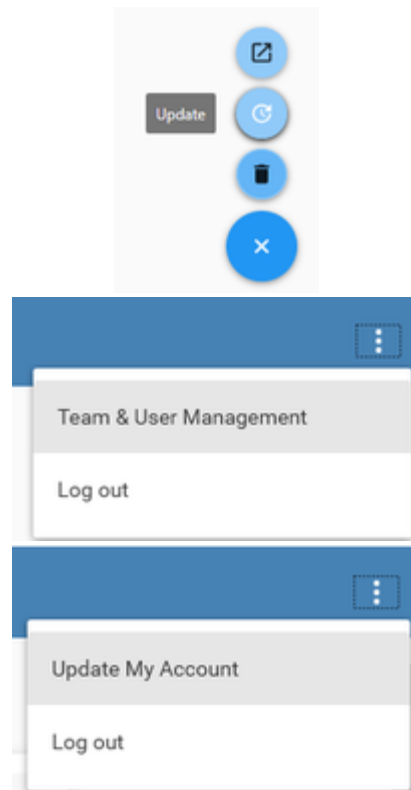


A side-by-side comparison of two screenshots of the "Simple Jira Application" interface. Both screenshots show a top navigation bar with "Simple Jira Application" and a dropdown arrow. Below the navigation bar is a sub-navigation bar with "My Account", "Team Management", and "User Management". The left screenshot shows the "My Account" view, with fields for "Login Name" (David), "First Name" (David), and "Last Name" (Test). The right screenshot shows the "Team Management" view, with fields for "Login Name" (Admin), "First Name" (Admin), and "Last Name" (Admin).



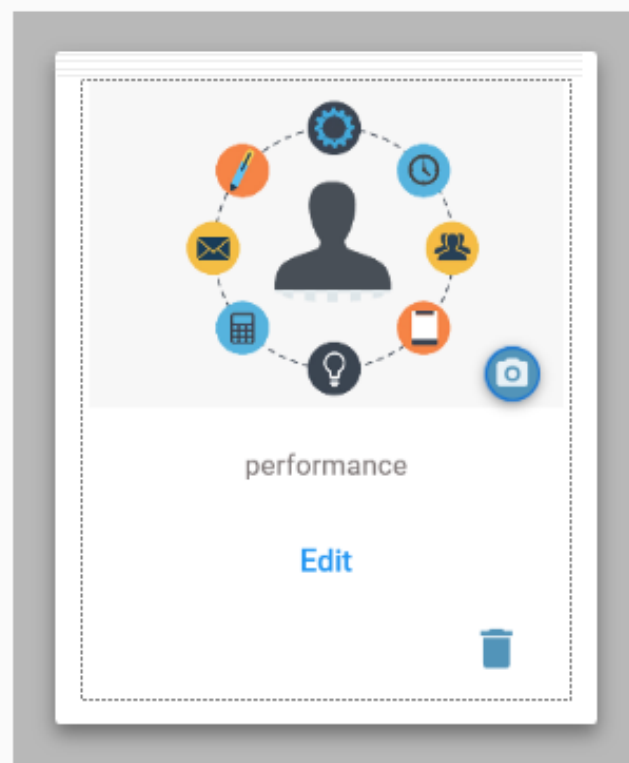
- Main Window:
  - resizable and popup function button
- ✓ Default project view with different popup boxes by different user



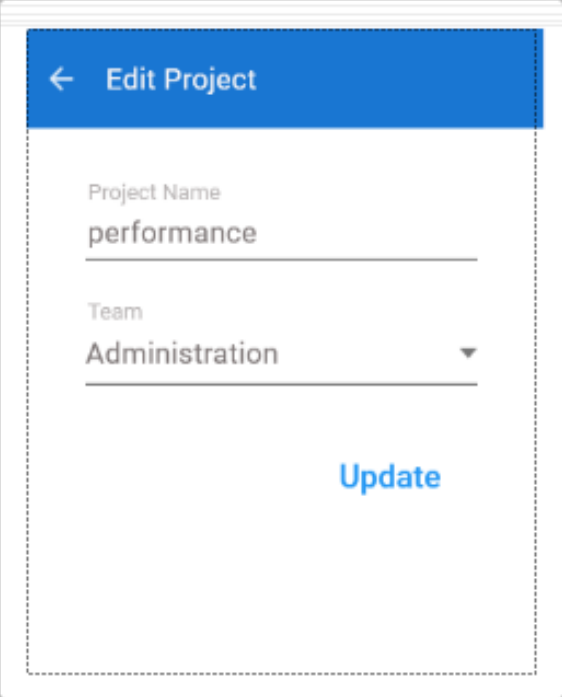


- Project View
    - Result set depends on login team
    - Flip Card with edit/delete function
- ✓ [Project card view and popup box](#)

## Projects Dashboard



## Projects Dashboard



The image shows a modal dialog box titled "Edit Project" with a blue header bar containing a back arrow and the title. The dialog has a white background with a dashed border. It contains two text input fields: "Project Name" with the value "performance" and "Team" with a dropdown menu showing "Administration". A blue "Update" button is located at the bottom right of the dialog.

← Edit Project

Project Name  
performance

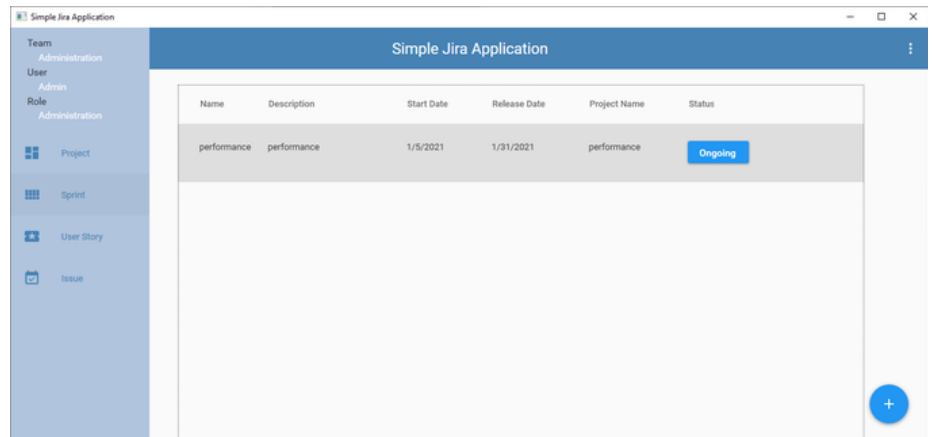
Team  
Administration ▼

Update

- Sprint View:

- List set depends login team, all sprints under the team's projects will be shown

✓ Sprint List View and detail dialog



The image shows a web application interface for "Simple Jira Application". It has a blue header bar with the title and a sidebar on the left with navigation links: Team, Administration, User, Admin, Role, Administration, Project, Sprint, User Story, and Issue. The main content area displays a table with columns: Name, Description, Start Date, Release Date, Project Name, and Status. The table contains one row with the following data: Name: performance, Description: performance, Start Date: 1/5/2021, Release Date: 1/31/2021, Project Name: performance, and Status: Ongoing. A blue button with a plus sign is located at the bottom right of the table.

Name	Description	Start Date	Release Date	Project Name	Status
performance	performance	1/5/2021	1/31/2021	performance	Ongoing

**Sprint** Update Sprint: performance

Sprint Name  
performance

Start Date  
2021-01-05

Release Date  
2021-01-31

Ongoing

performance

performance

Cancel Update

- User Story View:
  - list set depends on login team, all user stories under current team's project's all sprints

#### ▼ User Story List View and Detail Dialog

**Simple Jira Application**

Name	Description	Create Date	Complete Date	Point	Sprint	Owner	Status
performance	performance	1/6/2021	1/26/2021	99	performance	GOTEM	Todo

+

UserStory

Add New User Story

User Story Name

Start Date

Complete Date

Sprint Name

Owner Name

Description

Status

Cancel

Save

Todo

DEV

TEST

InValidation

Ready

Documenting

DONE

- Issue View:
  - List data depends on current team, completed date is set when status changed to resolved
  - Select image as attachment(1ps only)
  - Sortable data grid
  - Scrollable, resizable
  - directable to user stories

#### Issue List view and detail dialog

Simple Jira Application

Team

Release Team

User

David

Role

Senior Designer

Project

Sprint

User Story

Issue

Simple Jira Application

Issue Id	Name	Description	Start Date	Complete Date	Priority	Category	Owner	User Story	Status
18	lgkh	hlyghty ghukhykuk	1/9/2021		VeryHigh	Task	Defect Tester	Another user story	InProcess
16	gerag	vdftfdd vdftfdd vdftfdd vdftfdd	1/10/2021	1/17/2021	VeryHigh	Task	David	New requirement test 2023 ffe	Resolved
17	CVAD VF FD DVD	CSDD DVC CSDD DVC CSDD DVC CSDD DVC CSDD DVC	1/11/2021		Medium	Defect	David	User story for tester	Blocked
15	test	cdsabrfdba cdsabrfdba cdsabrfdba	1/15/2021		Low	Defect	Defect Tester	User story for tester	Resolved
21	hgkh	hghghh	1/20/2021		VeryLow	Defect	David	New requirement test 2023 ffe	InProcess



Issue

Add New Issue

Issue Category

Start Date

Issue Name

Complete Date

Description

Priority

Related User Story Name

Status

Owner

Add Photo

Cancel

Save

Issue

Update Task

Issue Category

Task

Start Date

2021-01-09

Issue Name

kjkhk

Complete Date

Description

hjkghj gkukykyuk

Priority

VeryHigh

Related User Story Name

Another user story


Status

InProcess

Owner

Defect Tester

Add Photo



Issue

Update Defect

Issue Category

Defect

Start Date

2021-01-20

Issue Name

hjkjhk

Complete Date

Description

hjkghjk

Priority

VeryLow

Related User Story Name

Another user story

Status

InProcess

Owner

Defect Tester

Add Photo

UserStory

Update Sprint: Another user

User Story Name

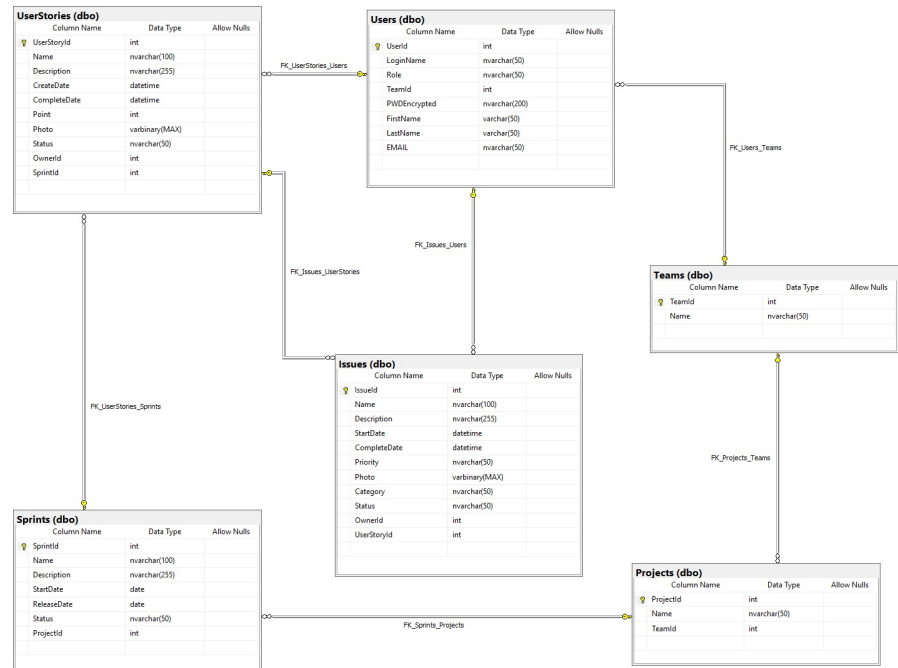
Another user story

Defect Tester

wdvwdvd

## Database Structure

### Database Structure



SimpleJiraDb\_Initial.sql

## Libraries

### Modern UI for WPF

#### Material Design In XAML and Dragablz for TabControl



**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 for better manual support

 Add namespace for material design and Dragablz in App.xaml

```
<Application x:Class="SimpleJiraProject.App"
             xmlns:materialDesign="
http://materialdesigninxaml.net/winfx/xaml
/themes"
             xmlns:dragablz="clr-namespace:
Dragablz;assembly=Dragablz"
             xmlns:local="clr-namespace:
SimpleJiraProject" >
```

 Add default resource dictionary and styles for all windows in App.xaml

```

<ResourceDictionary>
    <ResourceDictionary.
MergedDictionaries>
        <materialDesign:BundledTheme
BaseTheme="Light" PrimaryColor="DeepPurple"
SecondaryColor="Lime" />
        <ResourceDictionary Source="
pack://application:,,,/MaterialDesignThemes.
Wpf;component/Themes/MaterialDesignTheme.
Light.xaml" />
        <ResourceDictionary Source="
pack://application:,,,/MaterialDesignThemes.
Wpf;component/Themes/MaterialDesignTheme.
Defaults.xaml" />
        <ResourceDictionary Source="
pack://application:,,,/MaterialDesignColors;
component/Themes/Recommended/Primary
/MaterialDesignColor.Blue.xaml" />
        <ResourceDictionary Source="
pack://application:,,,/MaterialDesignColors;
component/Themes/Recommended/Accent
/MaterialDesignColor.Indigo.xaml" />
        <ResourceDictionary Source="
pack://application:,,,/Dragablz;component
/Themes/materialdesign.xaml"/>
    </ResourceDictionary.
MergedDictionaries>
    <Style TargetType="{x:Type
dragablz:TabablzControl}" BasedOn="
{StaticResource
MaterialDesignTabablzControlStyle}" />
</ResourceDictionary>

```



Add special component resource dictionary into each window XAML

```
<Window.Resources>
    <ResourceDictionary>
        <ResourceDictionary.
MergedDictionaries>
            <ResourceDictionary Source="
pack://application:,,,/MaterialDesignThemes.
Wpf;component/Themes/MaterialDesignTheme.
PopupBox.xaml" />
            <ResourceDictionary Source="
pack://application:,,,/MaterialDesignThemes.
Wpf;component/Themes/MaterialDesignTheme.
Button.xaml" />
        </ResourceDictionary.
MergedDictionaries>
    </ResourceDictionary>
</Window.Resources>
```



Add appropriate resource dictionary for the theme, use DynamicResource so they automatically update if the theme changes.

```

<Window x:Class="SimpleJiraProject.
MainWindow"
        xmlns="http://schemas.microsoft.com
/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com
/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com
/expression/blend/2008"
        xmlns:mc="http://schemas.
openxmlformats.org/markup-compatibility/2006"
        xmlns:materialDesign="
http://materialdesigninxaml.net/winfx/xaml
/themes"

        TextElement.Foreground="
{DynamicResource MaterialDesignBody}"
        TextElement.FontWeight="Regular"
        TextElement.FontSize="14"
        TextOptions.TextFormattingMode="
Ideal"
        TextOptions.TextRenderingMode="
Auto"
        Background="{DynamicResource
MaterialDesignPaper}"
        FontFamily="{DynamicResource
MaterialDesignFont}"

        xmlns:local="clr-namespace:
SimpleJiraProject"
        mc:Ignorable="d"
        WindowStartupLocation="CenterScreen"
        Foreground="White" Closed="MainWindow_Closed"
        Title="Simple Jira Application"
        MinHeight="600" MinWidth="1080" Height="
637.5" Width="1340" >

```



Use MaterialDesign style for different components

```

<materialDesign:PopupBox
        Style="{StaticResource
MaterialDesignMultiFloatingActionPopupBox}"
        <StackPanel>
                <Button ToolTip="
New" Click="btNew_Click"
                                Content="
{materialDesign:PackIcon Kind=OpenInNew,

```

```

Size=20}"/>

        <Button ToolTip="
Update" Click="btUpdate_Click"
                                Background="
{DynamicResource PrimaryHueLightBrush}"
                                Foreground="
{DynamicResource
PrimaryHueMidForegroundBrush}"
                                Content="
{materialDesign:PackIcon Kind=Update,
Size=20}"/>

        <Button ToolTip="
Delete" Click="btDelete_Click"
                                Content="
{materialDesign:PackIcon Kind=Delete,
Size=20}">

                                <Button.
Background>

                                <SolidColorBrush Color="{StaticResource
Primary300}"/>

                                </Button.
Background>

                                <Button.
Foreground>

                                <SolidColorBrush Color="{StaticResource
Primary300Foreground}"/>

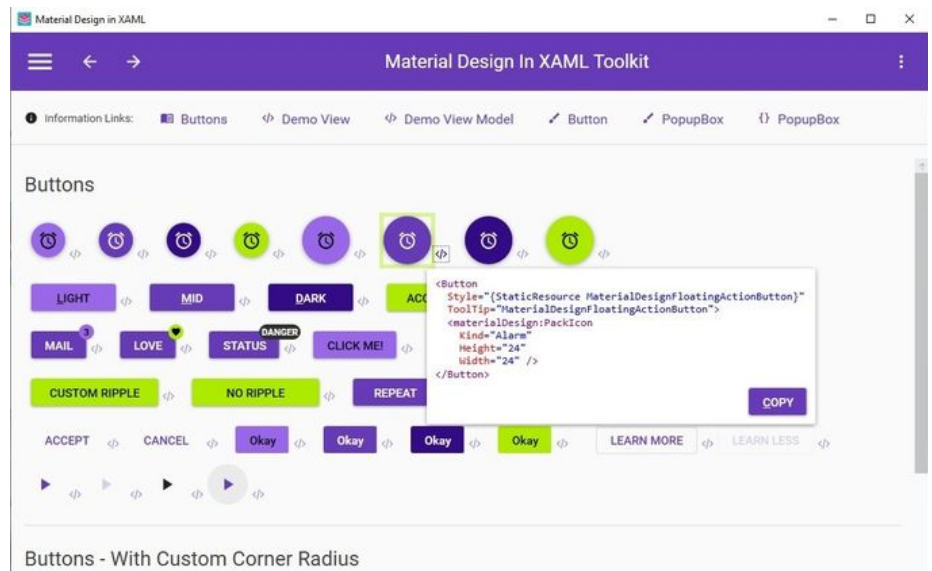
                                </Button.
Foreground>

                                </Button>
        </StackPanel>
    </materialDesign:PopupBox>
    <dragablz:TabablzControl Margin="
0,15, 0, 0" >
        <TabItem Header="My Account" x:
Name="tiMyAccount" />
        <TabItem Header="Team
Management" x:Name="tiTeam" />
    </dragablz:TabablzControl>

```



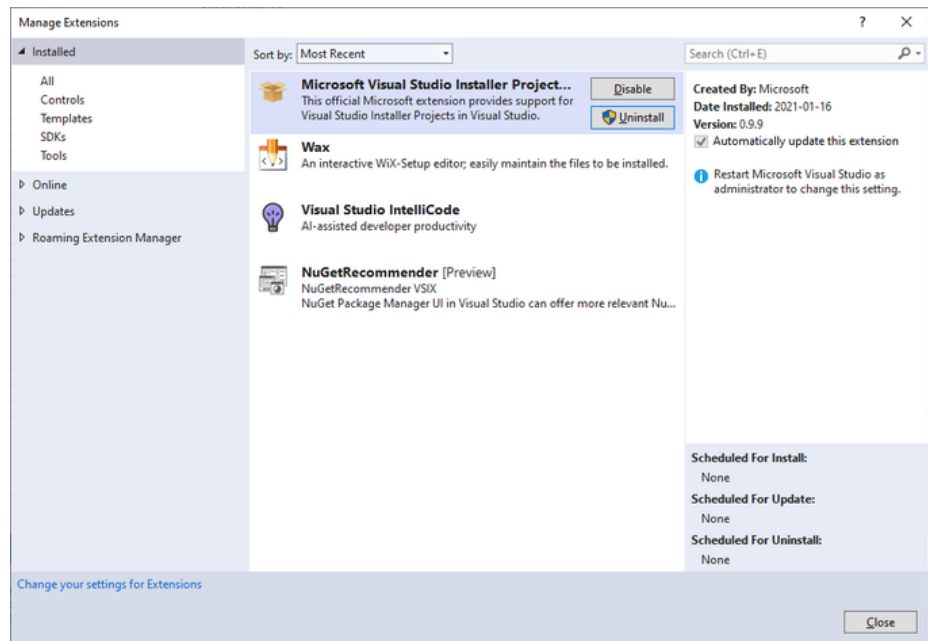
More usage according to the toolkit from github



## Material Design In Xaml Toolkit and Dragablz Tutorial

### Installer

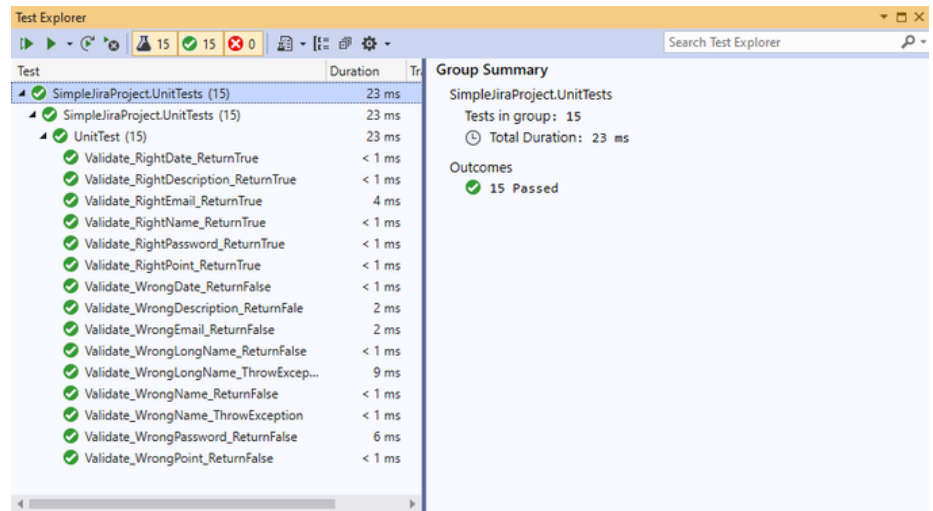
#### MS VS Installer project from extension



### MS Unit Tests

#### Unit Test for class GeneralValidation





Microsoft Azure Cloud Database

SQL Server Database - SimpleJiraDB (simplejira/SimpleJiraDB)

Database Storage Backup – Export from Azure and saved in Blob container(simplejiraexport)

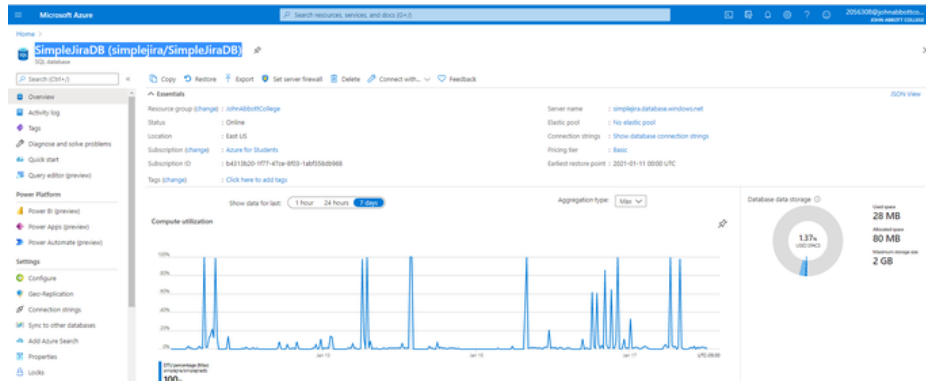
▼ Data structure scripts export from database by SSMS



▼ Database backup file(.BACPAC file) downloaded from Azure



## Database Overview

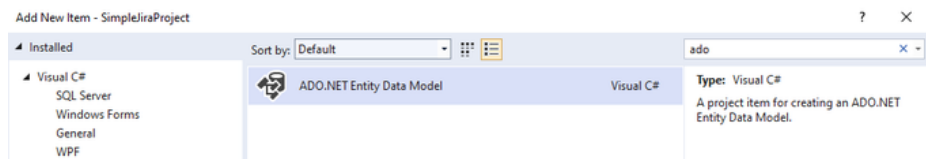


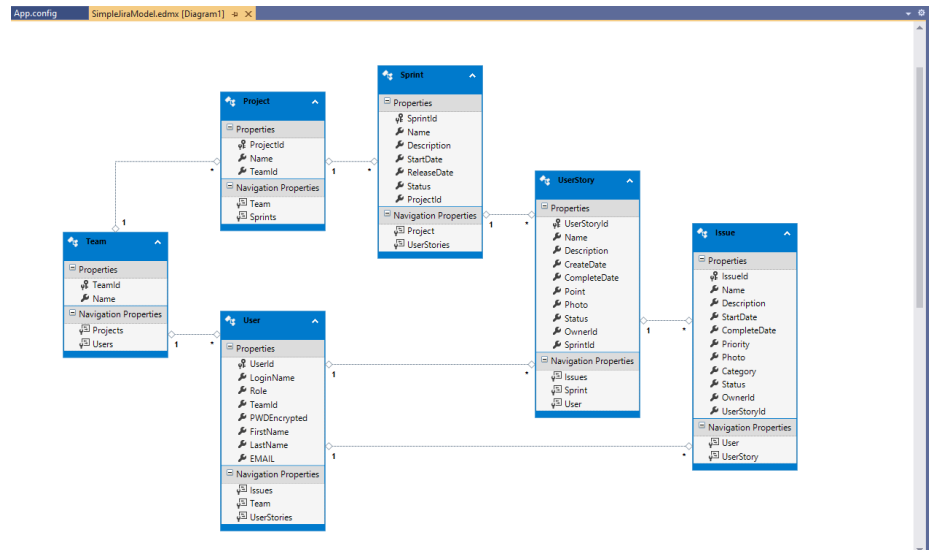
## Entity Framework - Database first

### App.config

```
<connectionStrings>
  <add name="SimpleJiraDBEntities"
    connectionString="metadata=res://*/
/SimpleJiraModel.csdl|res://*/
/SimpleJiraModel.ssdl|res://*/
/SimpleJiraModel.msl;provider=System.Data.
SqlClient;provider connection string="
data source=simplejira.database.windows.net;
initial catalog=SimpleJiraDB;persist
security info=True;user id=sqladmin;
MultipleActiveResultSets=True;
App=EntityFramework" providerName="
System.Data.EntityClient" />
</connectionStrings>
```

## Entity data model





## Major Problem & What we learned

### User Login / Logout

#### App.xaml

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

### SecurePassword by Advanced Encryption Standard (AES) Algorithm

#### Encrypt/Decrypt

#### C# – Encrypt/Decrypt Password

```

// Encrypt plain text
public static string Encrypt(string strData)
{
    return Convert.ToBase64String
(Encrypt(Encoding.UTF8.GetBytes(strData)));
}

// Create AesManaged, Encryptor,
MemoryStream, CryptoStream from MemoryStream
and Encrypter and write it.
public static byte[] Encrypt(byte[] strData)
{
    // Generate password, which will
be used to derive the key.
    PasswordDeriveBytes passbytes =
new PasswordDeriveBytes(strPermutation, new
byte[] { bytePermutation1,

bytePermutation2,

bytePermutation3,

bytePermutation4});
    // Create Aes that generates a
new key and initialization vector (IV).
    MemoryStream memstream = new
MemoryStream();
    Aes aes = new AesManaged();
    aes.Key = passbytes.GetBytes(aes.
KeySize / 8);
    aes.IV = passbytes.GetBytes(aes.
BlockSize / 8);

    CryptoStream cryptostream = new
CryptoStream(memstream, aes.
CreateEncryptor(), CryptoStreamMode.Write);
    cryptostream.Write(strData, 0,
strData.Length);
    cryptostream.Close();
    return memstream.ToArray();
}

//Call Encrypt
PWDEncrypted = SecurePassword.Encrypt
(tbConfirmPassword.Password);

```

## User Management

### ▼ Add team automatically if signup user input a new team

```
public int Team_Check(string team)
{
    Team chooseTeam = Globals.
simpleJiraDB.Teams.Where(t => t.Name.Equals
(team)).FirstOrDefault();
    int teamId = chooseTeam != null
? chooseTeam.TeamId : 0;
    if (teamId == 0)
    {
        Team newTeam = new Team {
Name = team };
        Globals.simpleJiraDB.Teams.
Add(newTeam);
        Globals.simpleJiraDB.
SaveChanges();
        return newTeam.TeamId;
    }
    else
    {
        return chooseTeam.TeamId;
    }
}
```

---

## Different view according to different team

### ▼ Filter data by LINQ from database

```
Globals.currentTeamProjectList = Globals.  
simpleJiraDB.Projects.Where(p => p.TeamId ==  
currentUser.TeamId).ToList<Project>();  
ProjectListView.ItemsSource = Globals.  
currentTeamProjectList;
```

```
Globals.currentTeamUserList = Globals.  
simpleJiraDB.Users.Where(u => u.TeamId ==  
currentUser.TeamId).ToList<User>();
```

```
IEnumerable<int> projectIds = Globals.  
currentTeamProjectList.Select(p => p.  
ProjectId).Distinct();
```

```
Globals.currentSprintList = Globals.  
simpleJiraDB.Sprints.Where(s => projectIds.  
Contains(s.ProjectId)).ToList<Sprint>();  
IEnumerable<int> sprintIds = Globals.  
currentSprintList.Select(sp => sp.SprintId).  
Distinct();
```

```
SprintListView.ItemsSource = Globals.  
currentSprintList;
```

```
Globals.currentUserStoryList = Globals.  
simpleJiraDB.UserStories.Where(us =>  
sprintIds.Contains(us.SprintId)).  
ToList<UserStory>();  
IEnumerable<int> userStoryIds = Globals.  
currentUserStoryList.Select(us => us.  
UserStoryId).Distinct();
```

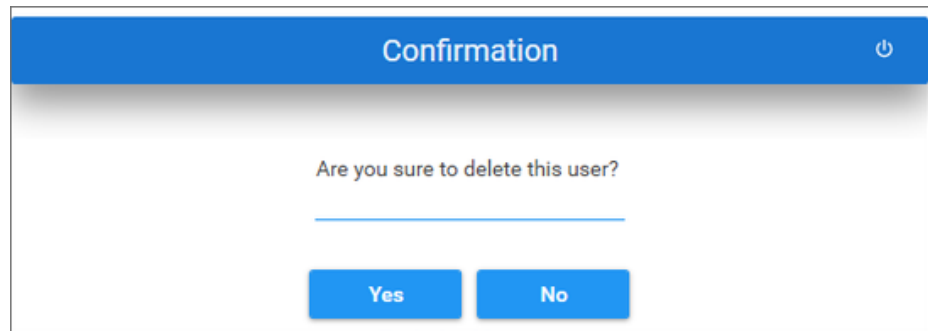
```
UserStoryListView.ItemsSource = Globals.  
currentUserStoryList;
```

```
Globals.currentIssueList = Globals.  
simpleJiraDB.Issues.Where(i => userStoryIds.  
Contains(i.UserStoryId)).ToList<Issue>();
```

---

#### Custom MessageBox

##### ▼ Custom MessageBox Dialog and usage



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

#### WPF Custom message box

#### Performance Issue

##### Large image loading from cloud database



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

```

IssueListView.ItemsSource = Globals.
simpleJiraDB.Issues.Select(x => new
IssueListItem
{
    IssueId = x.IssueId,
    Name = x.Name,
    Description = x.
Description,
    StartDate = x.StartDate,
    CompleteDate = x.
CompleteDate,
    Priority = x.Priority,
    Status = x.Status,
    Category = x.Category,
    OwnerId = x.OwnerId,
    UserStoryId = x.
UserStoryId,
    User = x.User,
    UserStory = x.UserStory
}).Where(iss => userStoryIds.
Contains(iss.UserStoryId)).ToList();
static long threshold =
2621440;    // 2.5MB
var fileLength = new
FileInfo(fileName).Length;

if(fileLength <
threshold)
{
    if (currentIssue
!= null)
    {
        currentIssue.
Photo = File.ReadAllBytes(fileName);
        image.Source
= new BitmapImage(new Uri(fileName));
    }
}

```

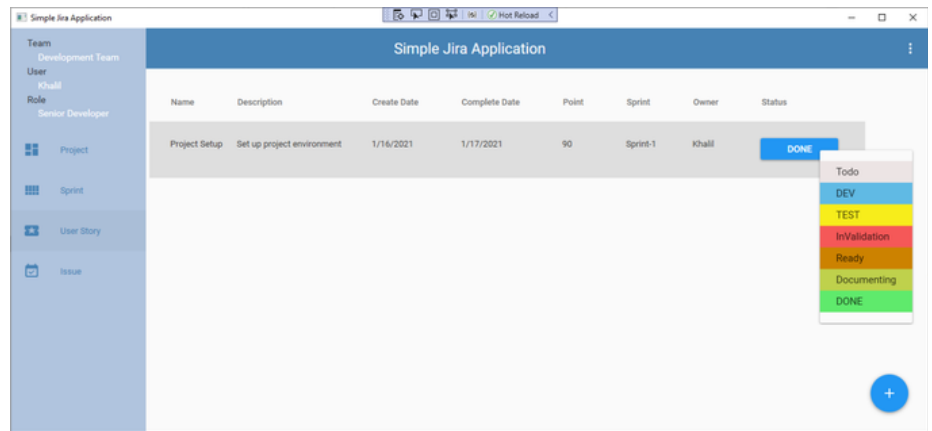
Status Update - color set and save change to database quickly

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

#### Challenges:

- Add item to list view
- Control the status





## Implement User Control

```
<GridViewColumn Header="Status" Width="130">
  <GridViewColumn.CellTemplate>
    <DataTemplate>
      <local:UserStoryStatus />
    </DataTemplate>
  </GridViewColumn.CellTemplate>
</GridViewColumn>
```

```
<UserControl x:Class="SimpleJiraProject.
UserStoryStatus"
             xmlns="http://schemas.microsoft.
com/winfx/2006/xaml/presentation"
             xmlns:x="http://schemas.
microsoft.com/winfx/2006/xaml"
             xmlns:mc="http://schemas.
openxmlformats.org/markup-compatibility
/2006"
             xmlns:d="http://schemas.
microsoft.com/expression/blend/2008"
             xmlns:local="clr-namespace:
SimpleJiraProject"
             mc:Ignorable="d"
             d:DesignHeight="450" d:
DesignWidth="800">
  <Grid>
    <Button Name="btStatus"
HorizontalContentAlignment="Center" ToolTip="
```

```

Right Click to update the Status" MinWidth="
115" >
    <TextBlock Name="tbStatus" Text="
{Binding Status , Mode=OneWay}" Height="18"
HorizontalAlignment="Center" />
    </Button>
    <Grid.ContextMenu>
        <ContextMenu>
            <MenuItem Header="Todo"
Name="miTodo" Background="#FFECE4E4" Click="
miTodo_Click"/>
            <MenuItem Header="DEV" Name="
miDEV" Background="#FF60BAE4" Click="
miDEV_Click"/>
            <MenuItem Header="TEST"
Name="miTEST" Background="#FFF7ED1B" Click="
miTEST_Click"/>
            <MenuItem Header="
InValidation" Name="miInValidation"
Background="#FFF55858" Click="
miInValidation_Click"/>
            <MenuItem Header="Ready"
Name="miReady" Background="#FFEA9500" Click="
miReady_Click"/>
            <MenuItem Header="
Documenting" Name="miDocumenting"
Background="#FFBFD14C" Click="
miDocumenting_Click"/>
            <MenuItem Header="DONE"
Name="miDONE" Background="#FF5FEA6C" Click="
miDONE_Click"/>

        </ContextMenu>
    </Grid.ContextMenu>
</Grid>
</UserControl>

```

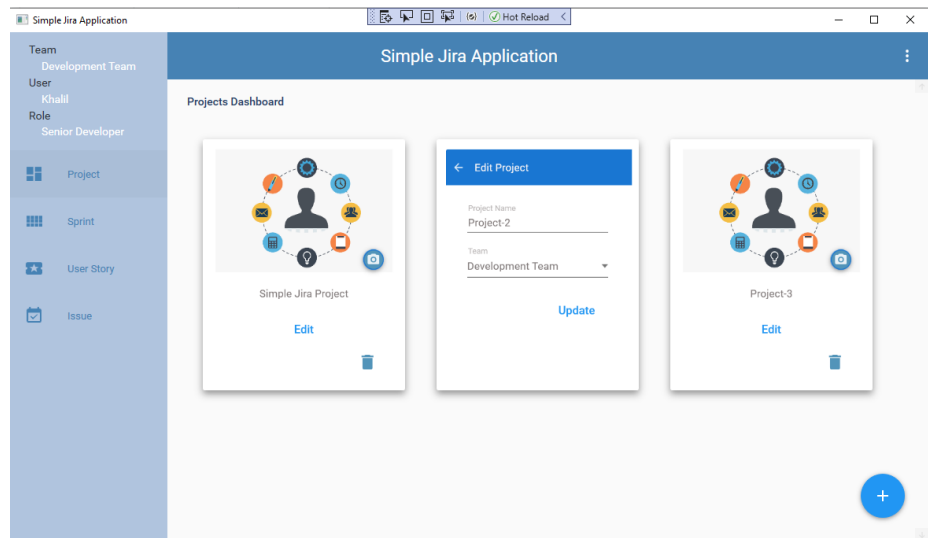
Project view - implemented by card rather than by list view

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



#### Challenges:

- Card Design
- Cards Creation
- Card Control



Using Item Control for cards creation



Implement User Control

```

<ItemsControl
Name="ProjectListView" x:FieldModifier="
public" >

<ItemsControl.ItemTemplate>

<DataTemplate>

<local:ProjectView />

</DataTemplate>

</ItemsControl.ItemTemplate>

<ItemsControl.ItemsPanel>

<ItemsPanelTemplate>

<WrapPanel/>

</ItemsPanelTemplate>

</ItemsControl.ItemsPanel>

</ItemsControl>

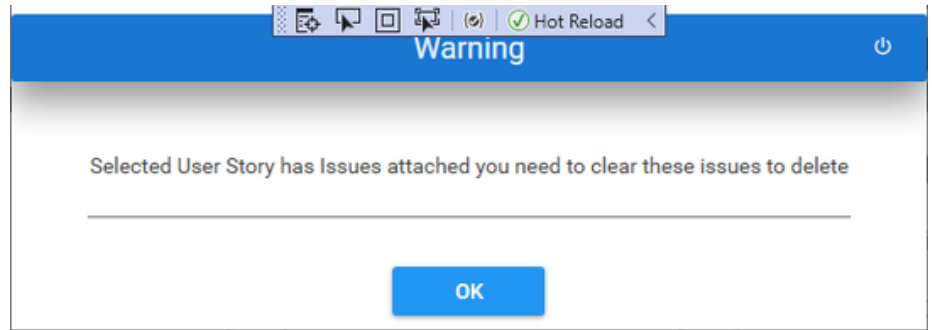
```

Delete Object - Handle delete related objects

▼ Handling the delete of objects

#### Challenges:

- Database constraint



Check the related objects before delete

#### Future Work

- Provide query or search function in each list of view or in main window to query globally
- Statistic chart generated for each list of view, especially for the user story page by using a library
- Add Project Themes
- Enhance the user Roles to optimize the Teams works
- UI/UX enhancement

#### Summary

- ✓ Design a 6 tables database with 1-Many relationships, practiced Microsoft SQL server DDL /DML statement, SSMS tools
- ✓ Use **C#** entity framework to create **WPF** project by database-first approach, implement key requirements for project
- ✓ Practice C# OOP skills and event handling
- ✓ Practice C#, WPF XAML
- ✓ Adapt Material Design UI library on WPF
- ✓ Create installer by MS VS installer project
- ✓ MS Unit Test for API testing