

Team 9-Project Proposal

Project Title: Simple Jira

Team Member: @ Khalil Hanna @ Ying LUO of Team9

Project Description:

- Create a simple version of the Jira which is a project resource management tool for Agile software development.
- To limit the project scale, initially only consider the task management, provide CRUD operation for project, sprint, user story and issues, logically sprint could be broke down by many user stories, and user stories can be related to many issues
- Key properties for each entity/object are sprint version, user story point, issue priorities, attached images for user story or task or defects, and different status for all entities, like Done, Ongoing, Blocked, etc..
- Owner assignment for each entity, teams assigned to sprint/project, users assigned to task/defect
- Main windows:
 - project list, update project name
 - sprint list of the current project team, with status if released/ongoing/planning, date for each milestone/stage
 - detail edit dialog to add new sprint, update sprint name, change sprint status, set sprint owner to team, set start and release data of sprint
 - user story list for the current sprint, with status of todo, ongoing, inprocess, completed
 - for backlog, it's still a user story but could be re-assigned to next sprint
 - update user story name, description, add/change photo, set/change owner of user story
 - start date is automatically set when it created
 - when issue's status change to done, the complete date will be set automatically
 - issue list linked to each user stories, like defect/task
 - detail edit dialog to add new issues, there are two categories could be chosen: defect/task, update owner of issues, change related to user stories.
- Main features
 - CRUD operations

Possible enhancement:

- user management, login and edit profile, including creator record for each user story/issue, authentication verification, etc...
- search feature by query keywords of task
- statistic chart or status report of user story and issues, how many in ongoing, done, released, inprocess, etc..
- comment by owners while changing status
- installer
- status or owner tracking log

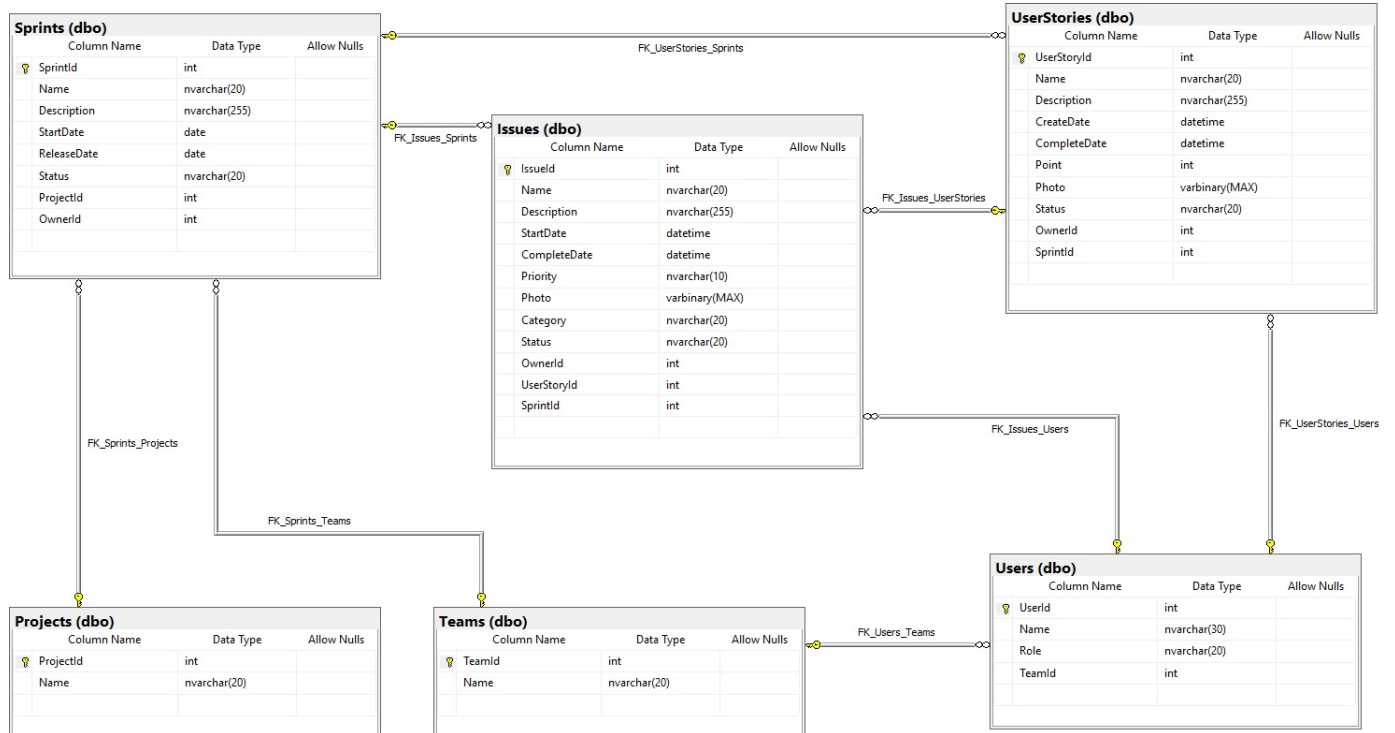
Time Management:

- Total 4 weeks project around
- Week1: Basic system design, UI and CRUD operations implementation
- Week2: NUnit Test, defect fix, incremental features of possible enhancement
- Week3: Keep sprint deliverables and incremental features
- Week4: Release delivery and presentation

ERD Diagram:

Initial database ERD, 4 tables at the beginning:

- Teams
- Users
- Projects
- Sprints
- UserStories
- Issues(including defect/task)



Mockups/Screenshots of Program Windows:

Sprint Backlog **Users**

Status: --

Description:

Start Date: yyyy-mm-dd

End Date: yyyy-mm-dd

Add Sprint **Edit Sprint** **Delete Sprint**

Add Sprint

Status:

Description:

Start Date:

Select a date

End Date:

Select a date

Save

Cancel

Status:

--

Sprint Details

Description:

Category:

--

Issue Owner:

--

Start Date:

yyyy-mm-dd

End Date:

yyyy-mm-dd

Create Issue

Edit Issue

Delete Issue

Add Issue 1*

Issue Name:

Status:

Description:

Category:

Issue Owner:

Start Date:

End Date:

546

24

Go to Sprints

Name:

Role:

Team:

Project Management:

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

Source Control:

[Bitbucket](#)

Knowledge Management:

[Confluence](#)

Technologies MUST use:

- WPF C#
- Azure SQL server database
- Entity Framework
- Multiple windows, dialogs
- NUnit Test
- Additional libraries
 - NUnit [NUnit Reference](#)
 - NLog ? [Nuget NLog](#)
 - Image processing library ?
 - Drawing Chart and PDF export library ? [Oxyplot](#)
 - Printing library ?

☒ As a developer, I need to create at least 10 NUnit test cases for project feature validation

☒ As a developer, I need to search and study which image processing library is the best suited for the project

Technologies may use:

- Suitable layout for each windows, preferred
 - GridSplitter [GridSplitter Tutorial](#) Considering adapt in defect/task descript/detail window to display image aside
 - TabControl [TabControl Tutorial](#) Considering adapt in main window to display each sprint detail information
 - ToolBar [ToolBar Tutorial](#) Considering adapt in defect/task edit dialog to modify the content
 - TreeView [Treeview Tutorial](#) Considering adapt in user story and linked defect/task view

☐ As a developer, I need to use Tab/TreeView for window layout design

- LINQ for query and search from database
- Ribbon Control: [Tutorial step by step](#) / [Simple Tutorial](#)

☒ As a developer, I need to use Ribbon control rather than menu in my project window