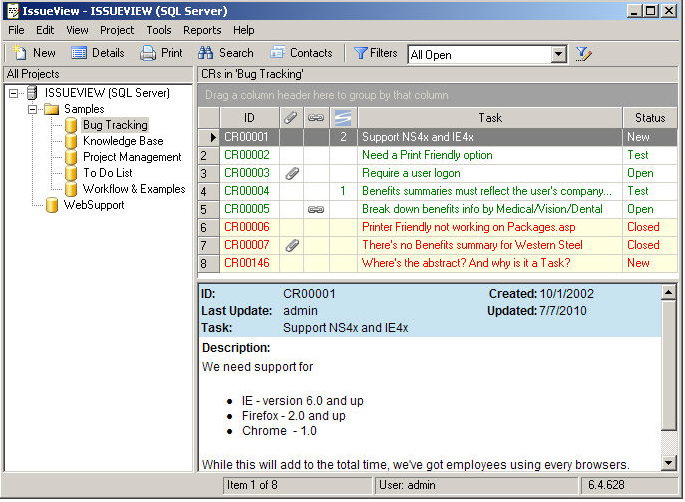
**Objective**

To reverse engineering an end-of-life software tool called IssueView which is a tracking system for tracking any kind of incident based data. For example, you can use IssueView to track software defects, customer requests, project tasks – anything where the basic unit to be tracked can be represented as an ‘Issue’.

We will re-implement the IssueView functionalities in C# using Visual Studio 2015 or 2017 and name the new program ‘IssueTracker’.

**IssueView Overview**

IssueView’s primary view is the Issue Explorer Window. Here you can select the project you wish to work with, and access detailed information about issues.



On the left side of the Issue Explorer is the Project Tree. The root node of the tree is the name of the IssueView Database. Here, we have SQL Server hosts the IssueView database. You can select the IssueView Project that you wish to work with by clicking on it in the tree. The Tree contains Projects (database icon) which contain issues, and folders (folder icon) which are organizational containers for projects. BTW, each project is implemented as a SQL table.

On the right side of the Issue Explorer is the issues Grid. All of the Issues of the currently selected project are displayed in the Grid. BTW, the issue is implemented as a row of the SQL table.

The Preview Pane (just below the Issue Grid) displays certain text fields of the currently selected issue.

The status bar, located at the bottom of the window, shows the current issue and count of issues, the UserID of the logged on user, and the version number of the program.

The button bar, located at the top of the window, contains buttons for the most common actions you’ll execute in IssueView (NEW, Details, Print, search, Contact, Filter, etc). Since we are not implementing the functionality of the button bar in the first phase, we will skip discussion on them.

**Implementation Approach**

IssueView is not a small application, therefore we will implement it in multiple phases. Recalling in any database application, **C**reate, **R**ead, **U**pdate and **D**elete (**CRUD**) are the four basic functions. In the first phase, we will only implement a subset of the **R**ead function. We can easily verify the **R**ead function of IssueTracker by running it against the database created by IssueView. To be compatible with the legacy database is one of the important goals of the porting effort.

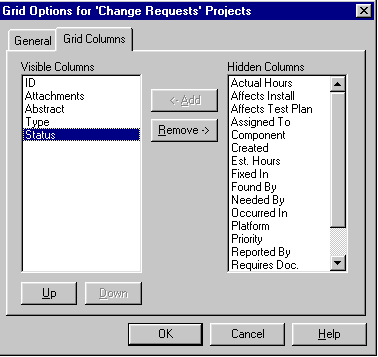
**Features of the first phase**

We will implement the main window including Project Tree, Issue Grid, and Preview Pane as well as their associated **R**ead operations:

* Display the project Tree on starting up and let user freely navigate the project tree
* Populate the Issue Grid when a project is selected and let user:
  1. Resize the Grid column by moving the mouse pointer over the column separator and shifting around
  2. Add or remove columns to/from the Grid via a Grid Column selection box (see below). Right click the Grid Column header will bring up a menu with ‘Grid Option’ on the list. Click on the ‘Grid Option’ will display a Grid Column selection box.
  3. A single click on any column header will sort the issue set by that column. A second single click reverses the sort order.
  4. Drag a column label to the gray area above will group issues by the column, multiple column grouping should be allowed.
  5. Color the row foreground/background. BTW, this is not a user interface operation, it will be done by the program (according to some rules) when loading data into the Grid.

Please note that #3 and #4 in red may need a third-party library to implement. We could defer them to phase 1.1, i.e., deliver phase 1 first, then a week or two later deliver phase 1.1. BTW, www.devexpress.com offers an award winning winforms library.

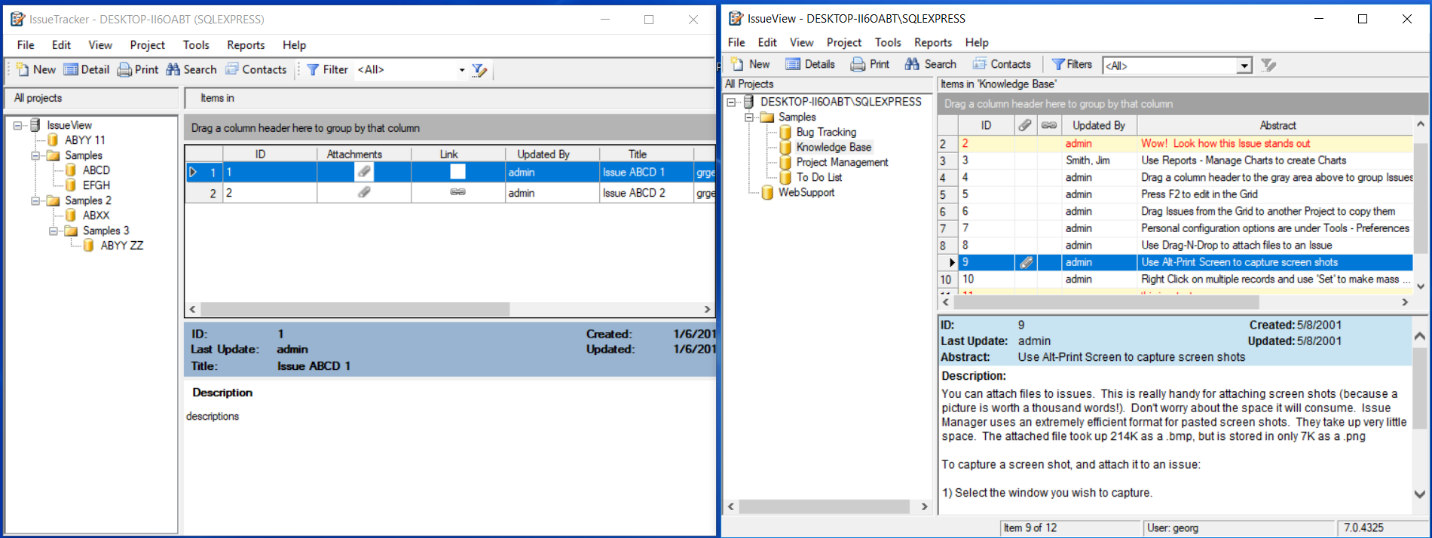
* Display some long field data to the preview pane when an issue in the Issue Grid is selected (clicked). The contents of these text fields are HTML tagged data, therefore we need a HTML viewer to render the data.



A prototype program has already been developed using C# to demonstrate the feasibility of the porting effort (see left picture below)

* The Project Tree is fully operational.
* The Issue Grid is very primitive. It can only display fixed number of columns.
* The Preview Pane is a standard textbox and it can’t render any HTML text.

The prototype program is also furnished for your reference. As can be seen, there are problems with the prototype such as missing parts and inferior graphics when compared to IssueView on the right.



Also, a design notes is also furnished to give more design related info.

A script ‘script.sql’ is also provided to generate all necessary tables including application data as well as tables reserved for program use internally to save you time. In phase 1 and phase 1.1, the program doesn’t have to generate or modify any tables, just simply to read existing tables to setup the project tree, populate the Grid, and fill up the preview pane, of course let user manipulate the Grid.

Please give your cost estimates for phase 1 and phase 1.1 for now. We will deal with each phase individually. Thank you!