High Level View

Figure 1 below pictures, at a high level, the new features for this assignment.
- Dashboard – A collection of stats on Issues.
- Record – A UI to document a new Issue.
- Modify – A UI to edit an existing Issue. It requires the user to select an existing issue for the current project. In the picture below, "Account Project" is the currently selected project.
- Remove – A UI to remove an existing Issue. It requires the user to select an existing issue for the current project. After selecting an issue to remove all that is needed is a confirmation (Yes/No) messagebox.
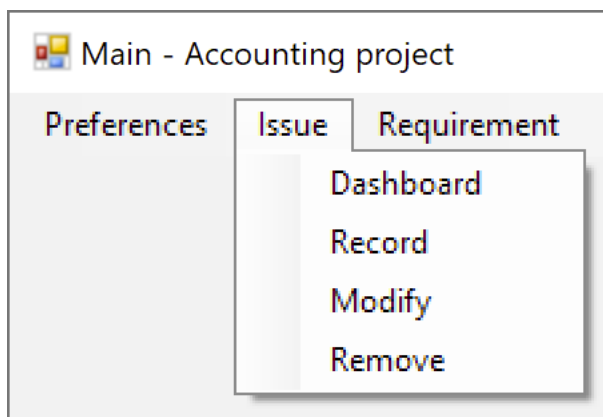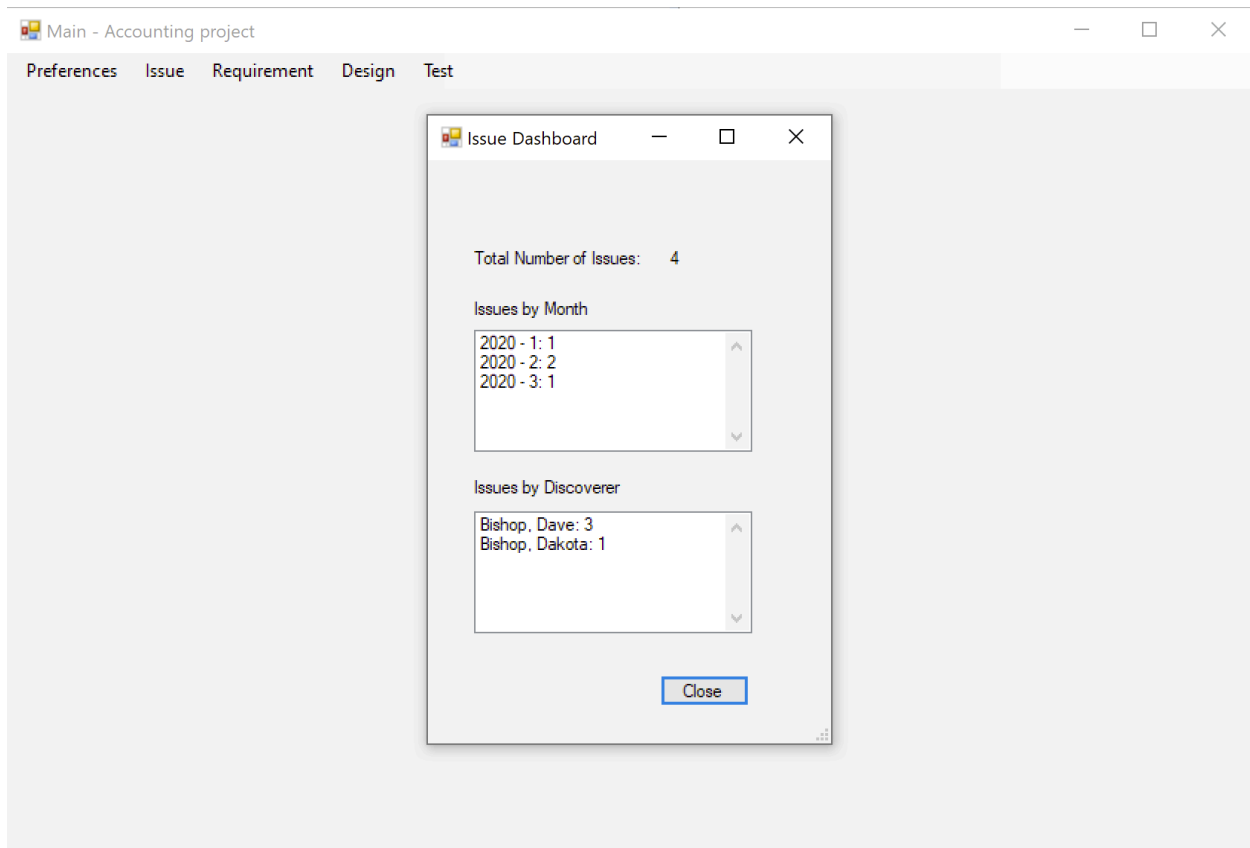

Figure 1 - New features for P6

Individual Feature View

Dashboard

There are three statistical elements to the dashboard.
1. The total number of issues. These are issues of any status, but only issues for the currently selected project.
2. Issues by month within year. This is a total of issues by the month and year based on the discovery date for each issue. One row per month with the total issues discovered for that month.
3. Issues by discoverer. One row per discoverer with the total number of issues discovered by that discoverer.

Record

1. The Id is system generated, unique and greater than any existing issue id. It is not editable or updatable.
2. The title is required and shall be unique across all projects.
3. A discovery date/time is required. The date/time shall not be in the future.
4. A discover is required. It shall be someone from the list of AppUsers
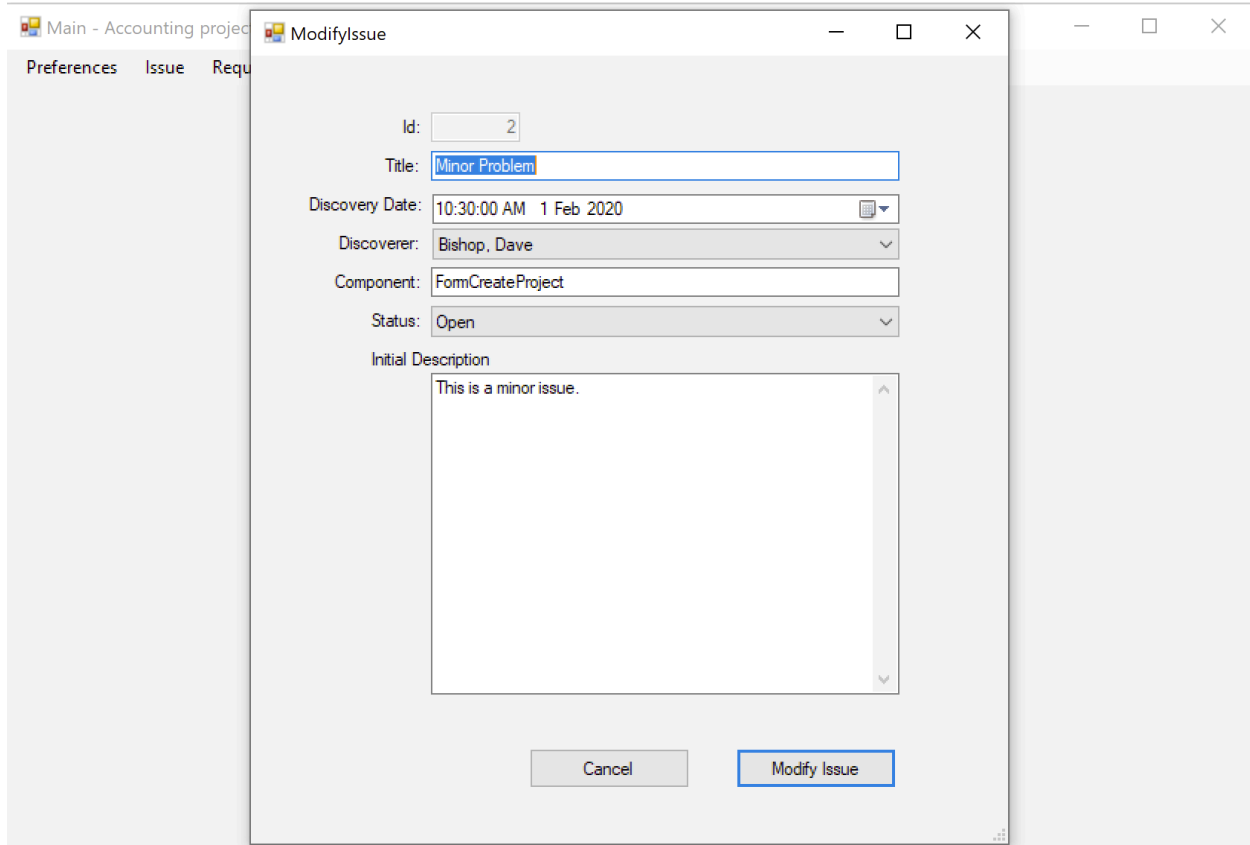5. A status shall be selected. It shall be from the FakeIssueStatusRepository. The valid issue status values are:

6. Initial Description. A multi-line textbox to document the issue and any repro steps if appropriate.

Modify

1. This list shall use a datagridview to display the existing issues for the currently selected project.
2. The columns shall be sized as shown.
3. The fields shall not updatable.
4. Any row can be selected.
5. If the Select Issue button is activated then the selected IssueId shall be set in a public attribute of the form.
6. If the Select Issue button is activated then the dialogresult shall be set to DialogResult.OK.
7. If the Cancel button is actived or the window is closed then the modify process shall be canceled and the user is returned to the main window.

When an issue has been selected and the Select Issue button activated.
1. The modify issue window shall be displayed.
2. The fields are pre-populated from the selected issue.
3. The title is required and shall be unique across all projects.
4. A discovery date/time is required. The date/time shall not be in the future.
5. A discover is required. It shall be someone from the list of AppUsers
6. A status shall be selected. It shall be from the FakeIssueStatusRepository. The valid issue status values are:

Remove

1. This list shall use a datagridview to display the existing issues for the currently selected project.
2. The columns shall be sized as shown.
3. The fields shall not be updatable.
4. Any row can be selected.
5. If the Select Issue button is activated then the selected IssueId shall be set in a public attribute of the form.
6. If the Select Issue button is activated then the dialogresult shall be set to DialogResult.OK.
7. If the Cancel button is actived or the window is closed then the modify process shall be canceled and the user is returned to the main window.
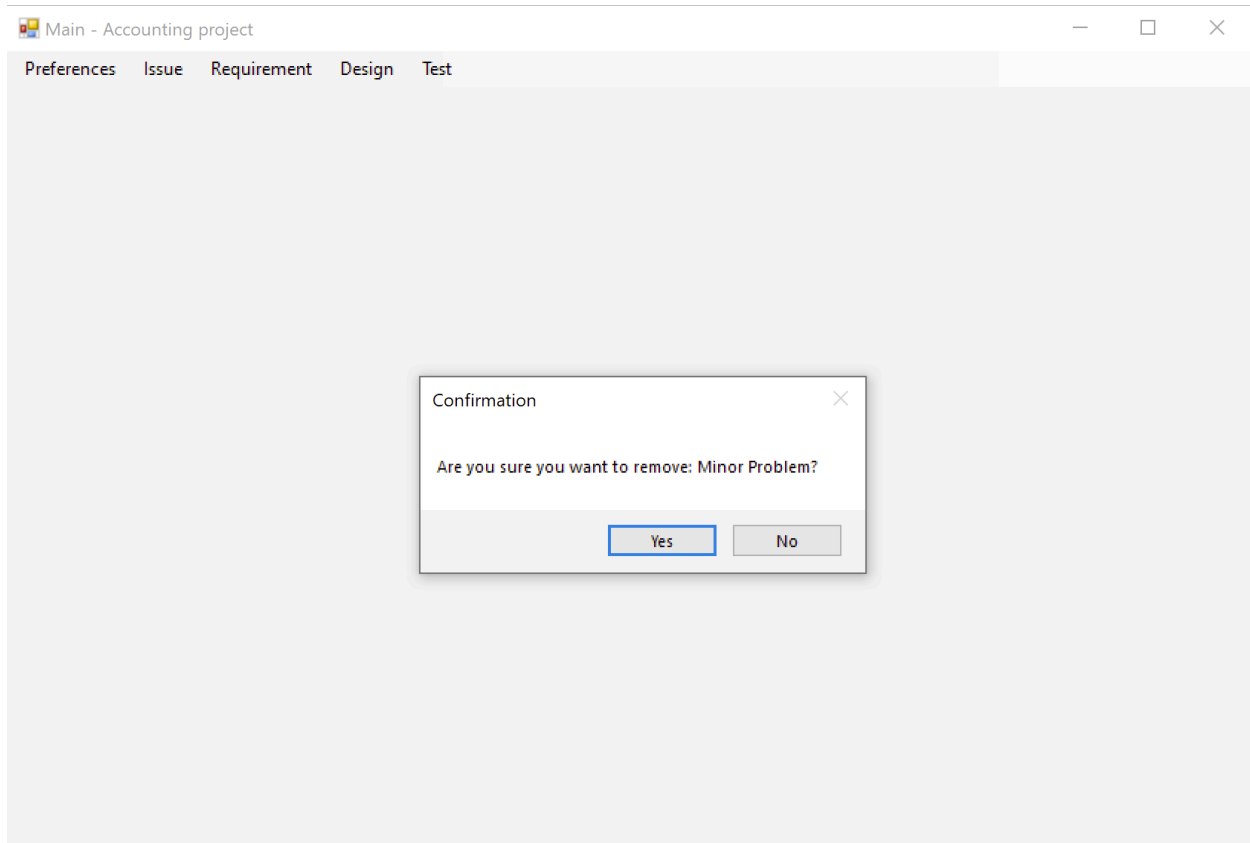
When an issue has been selected and the Select Issue button activated.
1. Display a message box with the phrase "Are you sure you want to remove: " concatenated with the title of the selected issue.
2. The message box shall have only Yes and No buttons
3. If the Yes button is activated then the selected issue shall be removed.
4. If the No button is activated then the remove process shall be stopped.

5. If the No button is activated the following message shall be displayed.
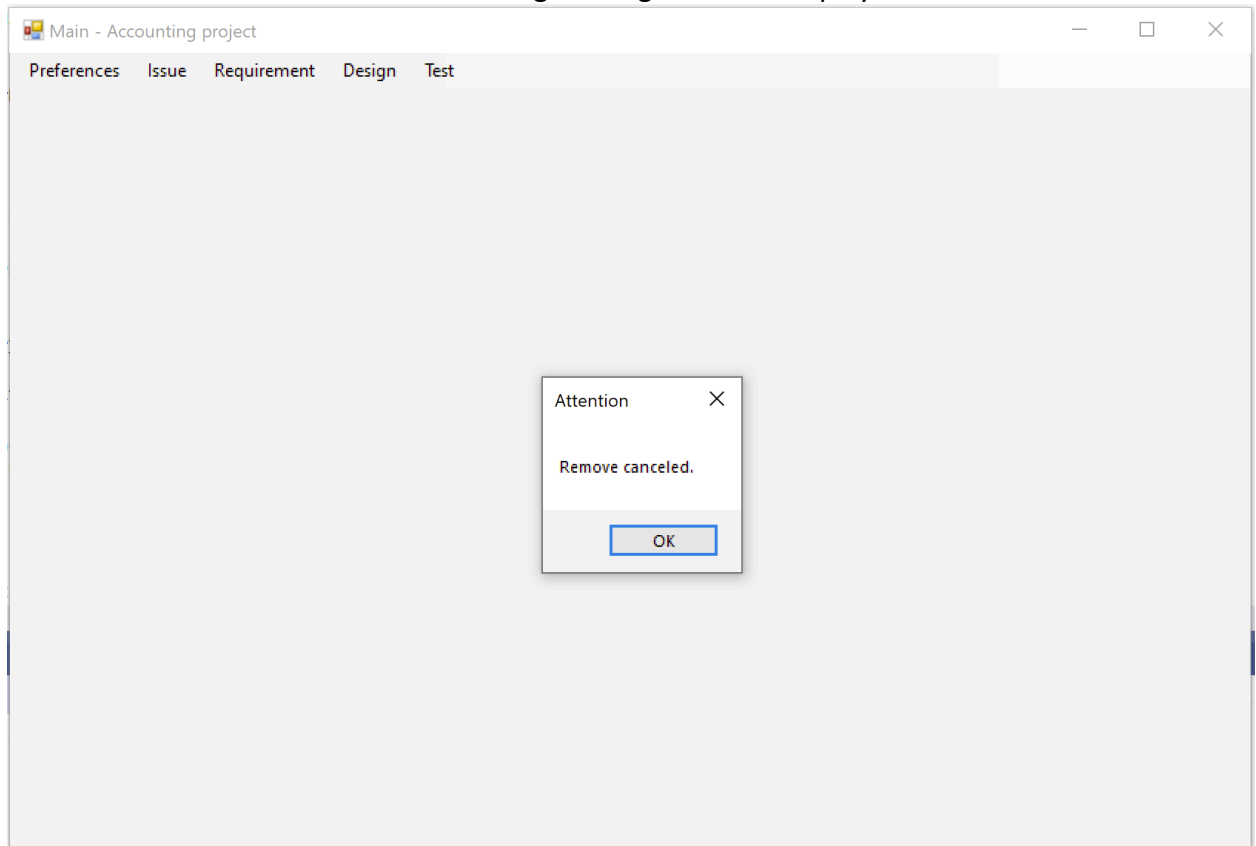
6. If the No button is activated then the user shall be returned to the main window.

UML Designs

Issue
+ Id : int
+ ProjectId : int
+ Title : string
+ DiscoveryDate : DateTime
+ Discoverer : string
+ InitialDescription : string
+ Component : string
+ IssueStatusId : int

<<IIssueRepository>>
+ Add(issue : Issue) : string
+ GetAll(ProjectId : int) : List<Issue>
+ Remove(issue : Issue) : bool
+ Modify(issue : Issue) : string
+ GetTotalNumberOfIssues(ProjectId : int) : int
+ GetIssuesByMonth(ProjectId : int) : List<string>
+ GetIssuesByDiscoverer(ProjectId : int) : List<string>
+ GetIssueById(Id : int) : Issue

FakeIssueRepository
+ NO_ERROR : string = ""
+ EMPTY_TITLE_ERROR : string = "A Title is required."
+ EMPTY_DISCOVERY_DATETIME_ERROR : string = "Must select a Discovery Date/Time."
+ FUTURE_DISCOVERY_DATETIME_ERROR : string = "Issues can't be from the future."
+ EMPTY_DISCOVERER_ERROR : string = "A Discoverer is required."
- Issues : List<Issue>
- ValidateIssue(issue : Issue) : string
- IsDuplicate(title : string) : bool

IssueStatus

+ Id : int
+ Value : string

<<IIssueStatusRepository>>

+ Add(Id : int, value : string) : void
+ GetAll() : List<IssueStatus>
+ GetIdByStatus(value : string) : int
+ GetValueById(Id : int) : string

FakeIssueStatusRepository

- IssueStatuses : List<IssueStatus>