Overview

A video overview: https://www.dropbox.com/s/48riyycjwjdfjvp/P4%20Overview.mp4?dl=0

This assignment will start a series of features to make an application that supports the software engineering process and the resulting artifacts. Two things happen at the start of the program. 1) The user must authenticate via a login window using the logic we created in previous assignments (see below). This is one of the key activities for this assignment. 2) Before the features like issue, requirements, etc. can be used a user must select a project (we will work on these in features future assignments).

Main form with login

When the program is executed the login window (Figure 1) is presented before anything other action can be initiated by the user. The password text box must not display the actual password. It must display the * character instead.

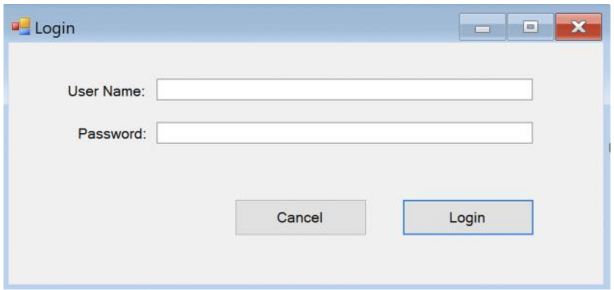

Figure 1 - Login Window

a. The application (FormMain and FormLogin) must use the class, interface and repository code from P3 to authenticate the user (Figure 2). If the user successfully authenticates, they can then access the menu on the main window (Figure 3). The menu items are just place holders for now.

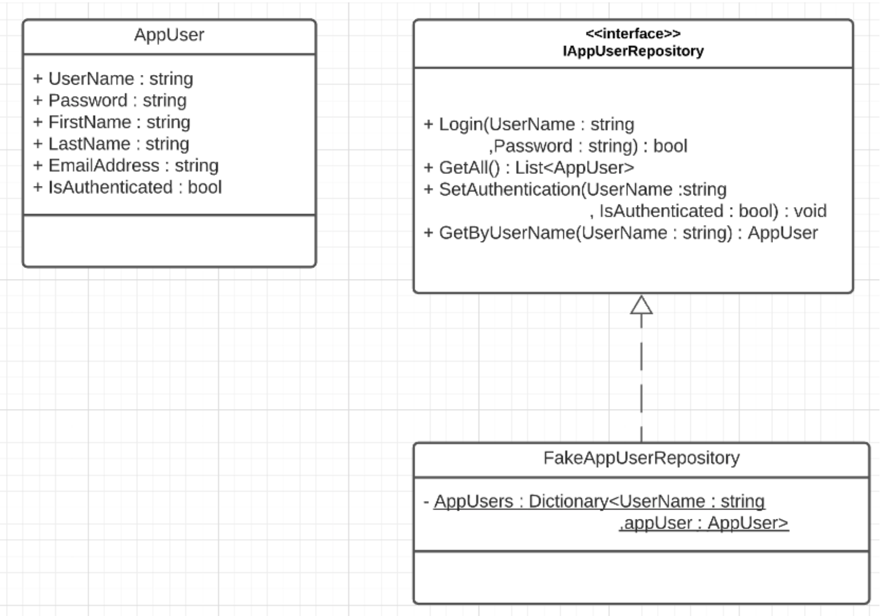
AppUser
+ UserName : string
+ Password : string
+ FirstName : string
+ LastName : string
+ EmailAddress : string
+ IsAuthenticated : bool
<<interface>>
IAppUserRepository
+ Login(UserName : string
,Password : string) : bool
+ GetAll() : List<AppUser>
+ SetAuthentication(UserName :string
, IsAuthenticated : bool) : void
+ GetByUserName(UserName : string) : AppUser
FakeAppUserRepository
- AppUsers : Dictionary<UserName : string
,appUser : AppUser>

Figure 2 - Class Diagram from P3

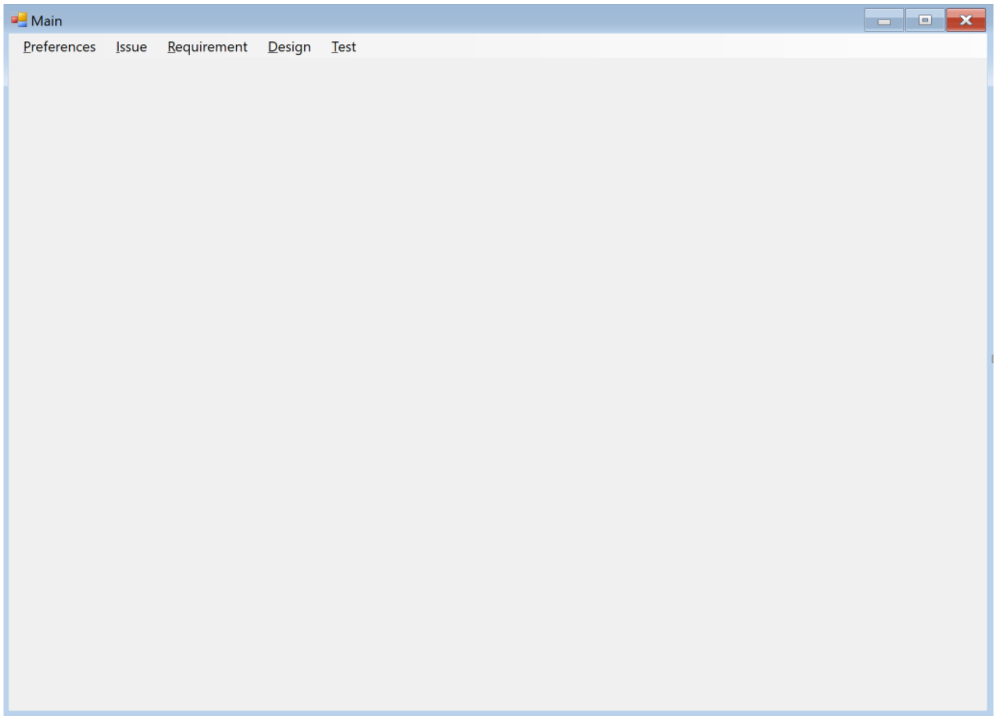

Figure 3 - Main Window

b. If the user fails to authenticate, the system must keep trying to authenticate or exit the application.

c. Menu Strip (No functionality at this point, just place holders)

   1) Menu strip with top level items of: Preferences, Issues, Requirements, Designs, Tests (Figure 3)

2) Preferences has sub items of: Select Project, Create Project, Modify Project, Remove Project (Figure 4)

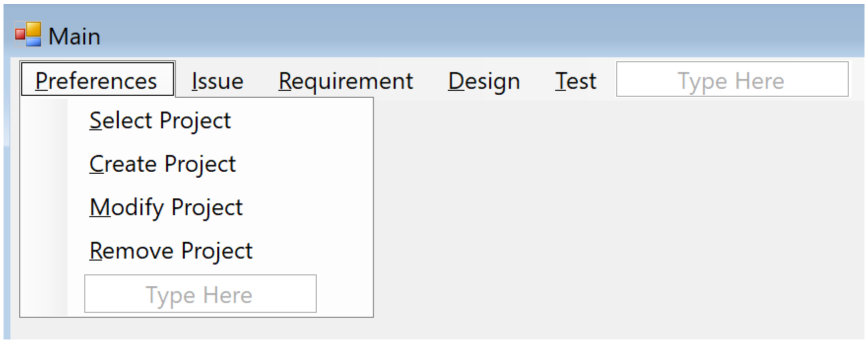

Figure 4 - Preferences Sub Items

3) Issues has sub items of: Dashboard, Record, Modify, Remove, Report (Figure 5)

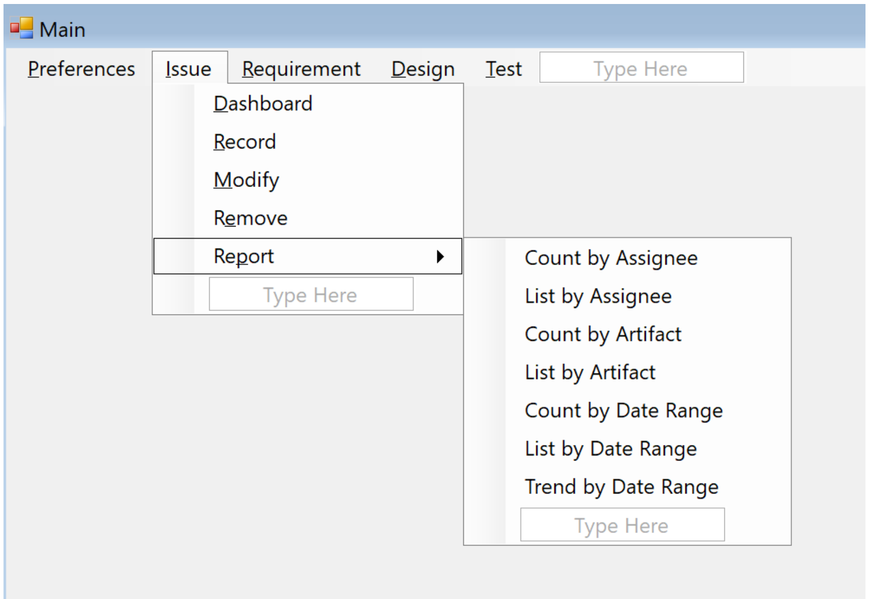

Figure 5 - Issues Menu Items

4) Report has sub items of: Count by Assignee, List by Assignee, Count by Artifact, List by Artifact, Count by Date Range, List by Date Range, Trend by Date Range (Figure 5)

The sequence diagram (Figure 6) outlines the design for the application at this stage.

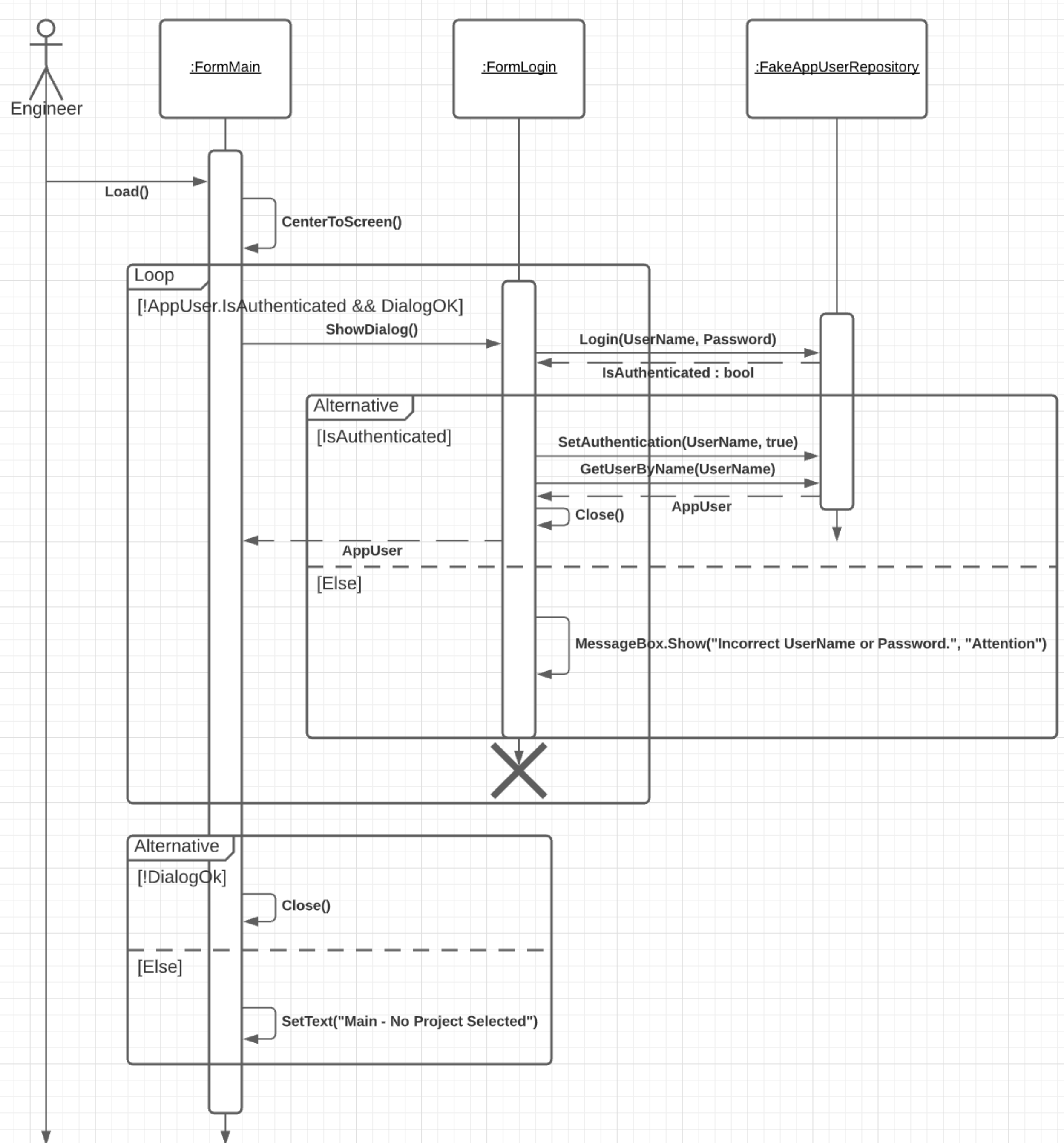

Figure 6 - Sequence Diagram

Grading

- Correct functionality and appearance per the requirements and design specified
- Evenly spread commits to GitHub across your team
- Clear, intentional and meaningful identifier names in all code
- Well-structured code
- Clear, intentional and meaningful use of GitHub (structure, commit comments, etc.)