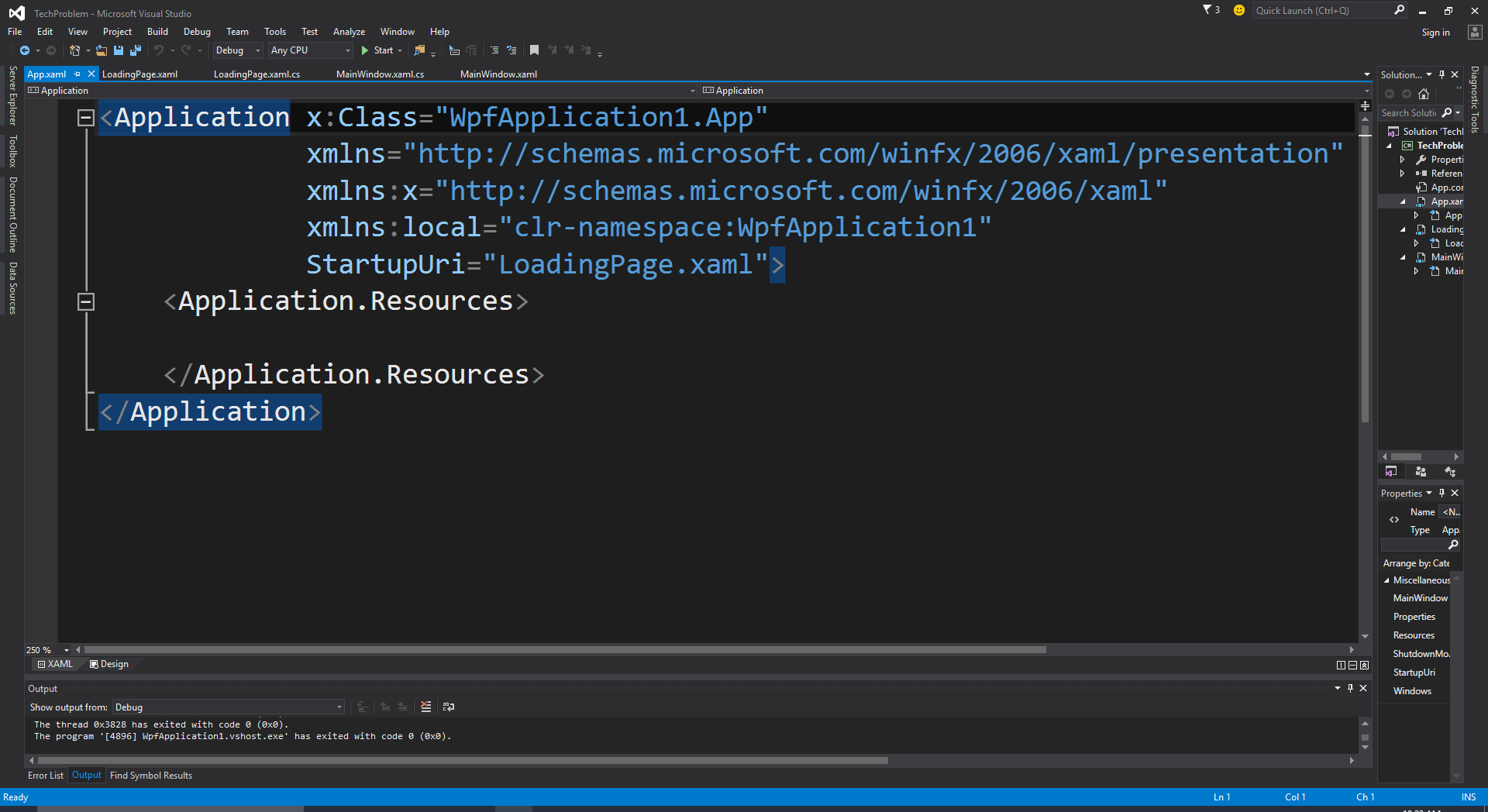
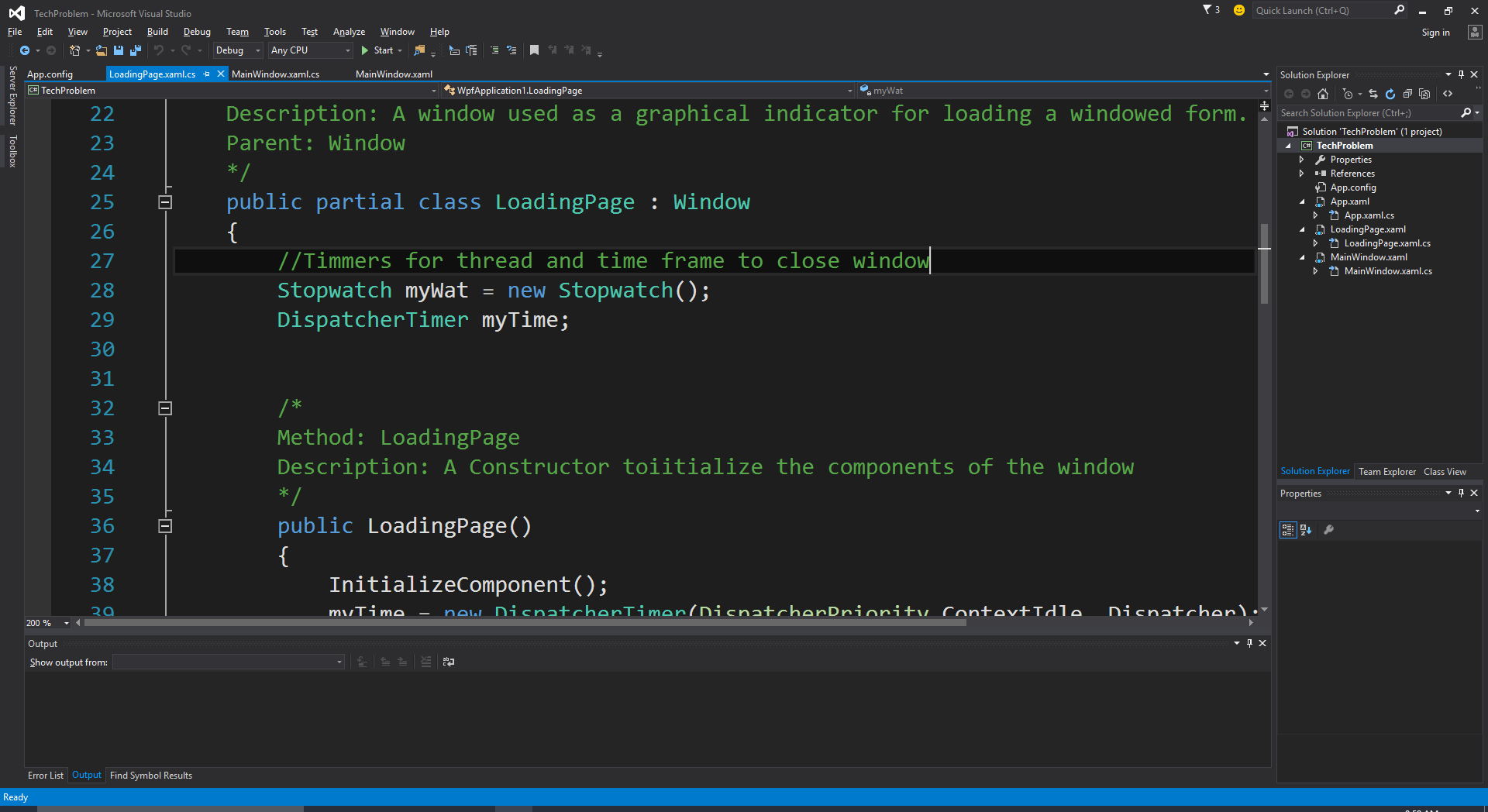
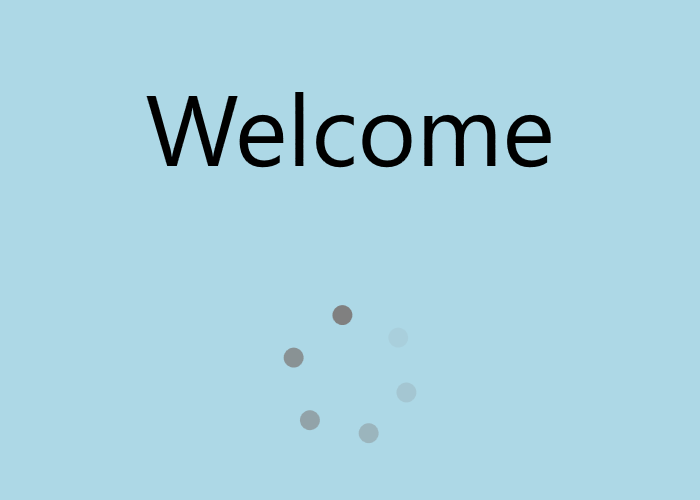
Technical Challenge Solutions

1. The WPF application must take 10 seconds to load.
2. While loading, the WPF application must display a landing page.

Both number one and number two were solved by creating a separate window form to serves as the loading page. This separate window was made to be invoked first and shown on screen for 10 seconds before it created an instance of the main WPF application. The *App.xaml* file was altered so StartupUri will have the value of “LoadingPage.xaml” which creates and shows the window that serves as the loading page. This will make it displayed first. 

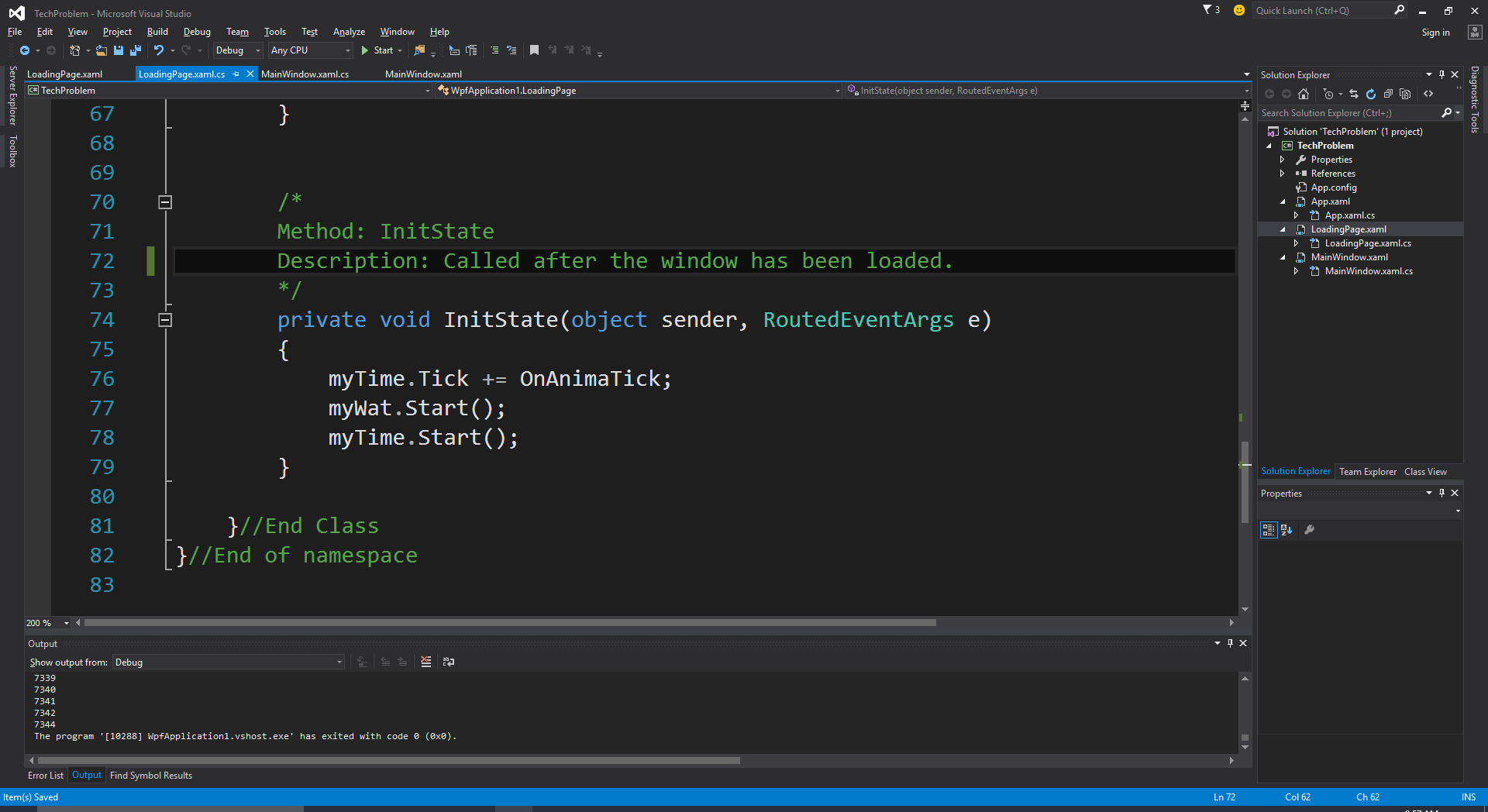
To make the loading page be shown and the WPF application to load for ten seconds, a Stopwatch instance called *myWat* was Implemented to keep track on how many milliseconds have passed by. If the amount of time exceeded 10000 milliseconds then the loading page was made to close and create the main WPF application. **This is explained more at the solutions to 9, 10, and 11.**

1. The Landing page must be a rectangle
2. The Length of the landing page must be between 5 and 8 inches.
3. The width of the landing page must be between 3 and 5 inches.
4. The landing page must include a title.
5. The landing page must include a spinner loading control.
6. The landing page must not have any controls that the user can interact with, e.g. buttons

*The LoadingPage.xaml file* was altered to include these UI elements to the loading page window. Particularly, the window was scaled to fit the desired range for its dimensions, it included a label control which acts as the title, and its WindowStyle was set to *“None”* so that it would not contain a title bar for users to interact with (Minimize, Restore Down, and Close). I also added a Canvas called cnvSpinner to hold Ellipses positioned to make a circular shape to represent the spinner loading control.

1. The spinner control must spin while the landing page is displayed.
2. Only the landing page can be displayed while the application is loading
3. Once the WPF application is finished loading, the landing page must be disposed and the main window must become visible.

When the LoadingPage is loaded, I invoked a Loaded Event called *InitState* in the *LoadingPage.xaml.cs* file. This event sets up the Tick for the DispatcherTimer object called *myTime* that is declared globally within the LoadingPage Class. It calls the Start method for *myTime* and also the Stopwatch object called *myWat* used to keep track of how many seconds have passed since the Loading Page was created.



Inside the *OnAnimaTick* event, is where the code that handles the spinning animation while the landing page is displayed. It rotates the Canvas.LayoutTransform called *SpinnerTransform* by an angle of 0.1 degrees per tick. This creates the animation for the spinner loading control. The if statement checks the time passed from the Loaded Event if it exceeds 10000ms (10s). If it has then the loading page is closed and the MainWindow myWin is shown on screen.

