**Requirements**

1. Must be a **web application** (not a desktop application) built in a “newer” web development framework
   1. .Net preferably – if you choose another framework, support from us will be limited
2. **Database** should preferably be in SQL server 2012 or above – if you choose another database option, support from us will be limited.
3. **Unit testing**
   1. 10% code coverage for unit tests. xUnit is a good framework for .NET.
4. **User Experience** Standards
   1. The application needs to be designed so that anyone, regardless of technical level, should be able to understand your website.
   2. Text should be clearly visible across desktop and mobile views
      1. This includes font sizes, proper background colors for fonts, font colors, etc.
   3. Use everyday language that users will be able to understand
      1. Ex. “2:00 PM” instead of “14:00:00”
      2. Ex. “Credit” instead of “CR”
      3. Spelling out the date and time in a readable format instead of a timestamp
   4. Text in tables need to be properly aligned for readability.
      1. Numeric needs to be right aligned.
      2. Alphanumeric needs to be left aligned.
   5. Pay attention to use of negative (or “white”) space in your design as well
5. **Front-End** Development
   1. Make the application responsive and aesthetically pleasing.
   2. Front-end framework/libraries are up to you but must be included in project (aka no external resources), but you must use at least one CSS framework such as Bootstrap (unless you want to make all the styles yourself).
   3. Use Commerce Bank styling.

Dark Green: #006747

Green: #4FA800

Blue: #007AA3

Yellow Orange: #FFB300

Red: #E30000

Fonts: Poppins for Headings, Open Sans for Body Text.

https://fonts.google.com/specimen/Poppins

https://fonts.google.com/specimen/Open+Sans

1. **Login Page**
   1. Simple login and passwords fields.
      1. Mask the password field.
      2. Password requirements:
         1. 8 characters minimum
         2. 1 upper case letter
         3. 1 symbol
         4. 1 number
   2. Login button
2. **Notification Rules** - Tool should allow for configurable notification rules to be created to notify user when transactions fit into a set of criteria.
   1. Users should be able to Add/Edit/Delete notification rules without technical assistance.
   2. These are examples (you have the liberty of coming up with what types of notification rules there are and how they are implemented):
      1. Minimum of 3 notification rules added to system
         1. Ex: Out of state transactions
         2. Ex: Timeframe usage
         3. Ex: Categories
3. **Notifications based on Notification Rules**
   1. Main Requirement: Notification when the user logs in
      1. Ex: Transaction in Alaska
      2. Ex: Transaction at 2AM
      3. Ex: Transaction from Spa
4. **Transaction Summary**
   1. Transaction list sorted by date
      1. Don’t need to worry about searching/filtering
   2. Users should have the ability to add transactions here, which should automatically trigger any associated notification rules
5. **Home Page** 
   1. Dashboard (Summary for triggered notification rules):
      1. Number of times each notification rule has been triggered over the past month and year
         1. Daily screen should be easy to read, easy to use, and provide a snapshot of data
      2. Ability to hide notification rules where the times tripped is zero
   2. Ability to pull/compare notification rule different timeframes
   3. Ability to export to spreadsheet

**Stretch Goals (Two are required)**

1. Deploy the project into a windows server/cloud instance
   1. Practice configuration properties for different environments
2. Create a Web API layer for backend interactions.
3. Use an open source reporting tool/business intelligence suite for all the reporting and its data visualization
4. Use pull requests/code review approved by a group member within your source control
5. Security scan your application and fix Critical issues at a minimum. OWASP ZAP is a good open source option
6. Session for remembering user if they close their browser and then log in again
7. Options for user if they forgot their username or password
8. Notifications via messaging center in the app
9. Notifications via email or text