Automatic screenshot capture (Prototype)

Background:

This initial prototype project has a single component:

1. Screen Capture Application – a Microsoft Windows application to automatically capture screenshots on a timer.

Requirements Placed Upon Developer:

- Developer MUST certify that all work is their own work and not subject to any intellectual property rights of any other person or entity;
- Developer must assign all rights for developed work to the Client, including but not limited to, the right to sell or otherwise license work to other party or parties and further develop said work.

Details on Screen Capture Application ("Screen Capturer"):

- See Mock-up for UI.
- Microsoft Windows Application (x86/32-bit).

Program Execution:

1. On open, program will identify separate monitors/displays connected to computer and load these in to drop-down selector 'Screen to Capture';
2. Program will load remaining default settings from configuration file:
    a. Default screen capture timer
    b. 'This Device' name
    c. Local copy directory path – default path is "My Documents\Screen Capturer\"
3. Program will initiate memory objects required to operate – key are two memory objects to store screenshots – 'previous screenshot' and 'current screenshot'.
4. When 'Capture On/Off' button is pressed (to turn on), the following actions are taken:
    a. A new directory is created within the Local copy directory (from config file) with name 'YYYY-MM-DD_HH-MM-SS' (date and time of when button pushed, using local computer's system date-time). If directory already exists of same name, display pop-up error message and don't proceed.
    b. Change Capture Button style to show a pushed in style of button
    c. Capture first screenshot (do not include mouse pointer in screenshot) – this screenshot is stored in the 'previous screenshot' memory object.
    d. Screenshot is written in PNG file format to the Local Copy directory path (sub-directory created in point 3a above), with filename of '<This Device Name> Screenshot YYYY-MM-DD_HH-MM-SS-SSS.png'. Where date-time (including millisecond) is from local computer's system date-time.
    e. Start Timer.
5. Whilst Capture button is still 'On' – every x milliseconds (as per 'Capture Screenshot' timer setting):
    a. Capture screenshot (do not include mouse pointer in screenshot) – this screenshot is stored in the 'current screenshot' memory object (overwriting old current screenshot).

   b. Compare ‘current screenshot’ with ‘previous screenshot’ – if different, do the following:
      i. Write screenshot in PNG file format to the Local Copy directory path (sub-directory created in point 3a above), with filename of ‘<This Device Name> Screenshot YYYY-MM-DD_HH-MM-SS-SSS.png’.
      ii. Copy ‘current screenshot’ in to ‘previous screenshot’ memory object (overwriting old previous screenshot)
   c. If NOT different (current screenshot is the same as previous screenshot), do nothing.
6. If ‘Pause’ button is pressed (to turn it ‘on’):
   a. Suspend timer/screenshot capture
   b. Change pause button style to show a pushed in style of button.
7. If ‘Pause’ button is pressed (to turn it ‘off’):
   a. Change pause button style to show a not-pushed in style of button.
   b. Resume timer/screenshot capture
8. When Capture button is pushed to turn it off:
   a. Stop timers/screenshot capture
   b. Change Capture button style to show a not-pushed in style of button.