**A purple rectangle with yellow text

Description automatically generated**

**Smart Snapshot**

A black camera with a yellow dot

Description automatically generated

**Project Team Members –**

Alan Koshy Luke,

Vaishak B Naik

**Project Overview**

The Smart Snapshot Application is a Java-based desktop application designed to facilitate the capture and management of screenshots. The application allows users to take screenshots at specified intervals or on-demand, view captured images, and export them to PDF format. The user interface is built using Swing, providing a responsive and intuitive experience for users.

**Objectives**

* The Smart Snapshot application aims to enable the users to capture screenshots effortlessly, either as set intervals or on-demand and allows users to customize their screenshot experience based on their specific needs allowing the users to capture exactly what they want when they want it.
* The application provides a user-friendly interface that simplifies the process of managing and previewing captured screenshots. With clearly labelled buttons, the users can navigate the application with ease.
* It also allows users to select multiple screenshots and export them in a convenient PDF format.
* The snipping tool provided by windows requires users to manually initiate each capture, which can be time consuming and inefficient for tasks requiring multiple screenshots. In contrast, the Smart Snapshot application allows for scheduled and manual captures, providing a solution for users who need to capture images frequently or in bulk

**Scope of the project**

The Smart SnapShot Application includes the following features:

**Screenshot Capture**: Users can capture screenshots manually or at defined time intervals.

**Preview Functionality**: A preview window displays captured screenshots, allowing users to review them.

**Export Option**: Users can select multiple screenshots and export them as a PDF document.

**User Preferences**: Users can set preferences for the screenshot interval and storage location.

**Functional Requirements**

|  |  |
| --- | --- |
| **Category** | **Functional Requirement** |
| User Interface | 1. **Buttons bar** containing |
|  | * **Start button** to initiate screenshot capture |
| * **Stop button** to terminate screenshot process |
| * **Timer button** to select screenshot mode(timer interval/ user demand) |
| 1. **Preview window** which displays the screenshots captured. It contains, |
| * A **Select All** button which would select all the screenshots on the window |
| * An **Export to PDF** button upon clicking would convert all the selected screenshots into a PDF. |
| **Screenshot Capture** | The application must allow users to capture screenshots in two modes: |
|  | * **User Demand Mode**: Users can manually trigger screenshot captures by clicking a button. |
| * **Timer Mode**: Users can set a timer that automatically captures screenshots at specified intervals. |
|  | * The application must provide visual feedback (e.g., flash effect) when a screenshot is taken. |
| **Folder Selection** | * The application must allow users to select a folder where captured screenshots will be saved. |
| * If a folder is already populated with screenshots, the application should prompt the user to confirm whether to delete existing files. |
| **Image Management** | * The application must store captured screenshots in the selected folder. |
| * The application must allow users to view captured screenshots in the Preview Window, with each image accompanied by a checkbox for selection. |
| **Export Functionality** | * The application must allow users to select multiple screenshots for export. |
| * Users must be able to export selected screenshots as a PDF document. |
| * The application must prompt users to enter a heading for the PDF before export. |
| **Error Handling** | The application must provide appropriate error messages for various failure scenarios, such as, |
| * Failure to capture the screen. |
| * Failure to save a screenshot. |
|  | * No folder selected for saving screenshots. |
| * No images available for export. |
| **Dynamic Updates** | * The Preview Window must dynamically update to reflect newly captured screenshots. |
| * The application must allow users to select or deselect images in the Preview Window, affecting the export functionality. |
| **User Preferences** | * The application must allow users to configure the screenshot interval through the Timer Button. |
| **Application Lifecycle** | * The application must handle opening, closing, and minimizing the main window appropriately. |
| * The application must allow users to exit gracefully, ensuring that all resources are released properly. |

**Technical Specification**

1. Class Diagram

A diagram of a computer

Description automatically generated

1. Flow Chart

A diagram of a company

Description automatically generated

A diagram of a flowchart

Description automatically generated

1. A screenshot of a computer program

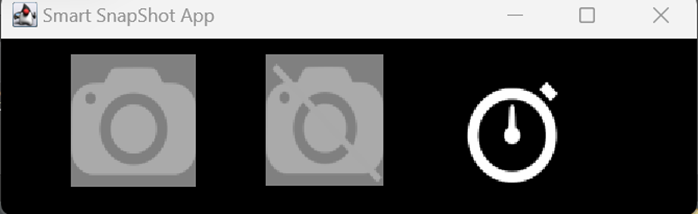
   Description automatically generatedSequence Diagram

A screenshot of a computer program

Description automatically generated

**User Guide**

Step 1.

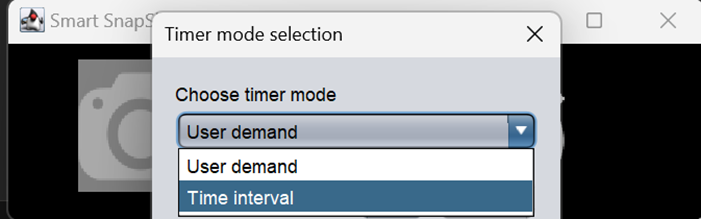


The application has 3 buttons:

1. Start Button
2. Stop Button
3. Timer Button

The Start Button & Stop Button is disabled until you select Timer Button.

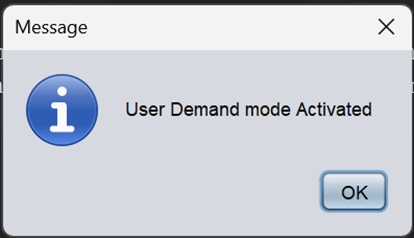
Step 2.



Timer Button has 2 options:

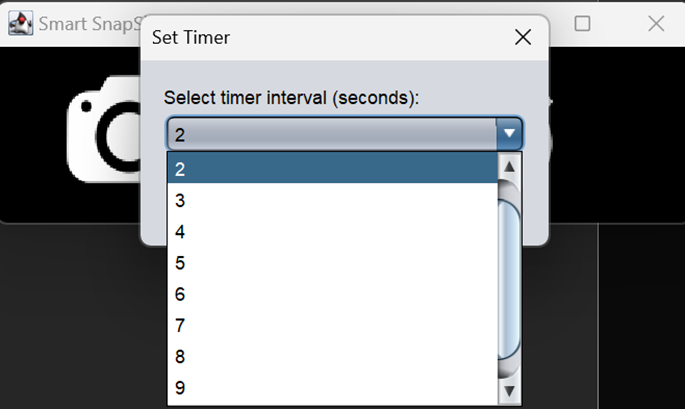
1. User Demand: Where user can take the screenshot based on their requirement.
2. Time Interval: Where for every selected time interval screenshot will be taken automatically.

Step 2.1: User Demand mode is selected

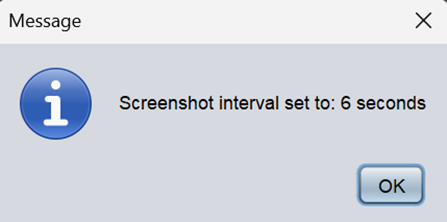


A popup window will be added to the screen.

Step 2.2: Timer Mode is selected



A List of intervals will be displayed, and users can choose the interval on their respective needs.

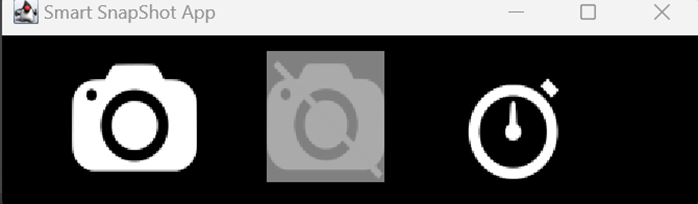


Based on the user interval selection the popup window will display the message of the time interval selected by the user.

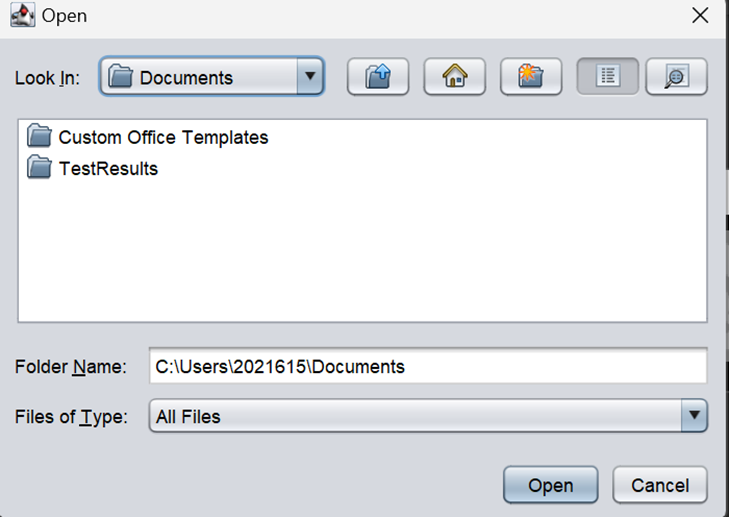
Step 3: Start Button will be enabled

“Start Button” will be enabled only after selecting the “Timer Interval” button,

The window will be minimized before taking the screenshot.

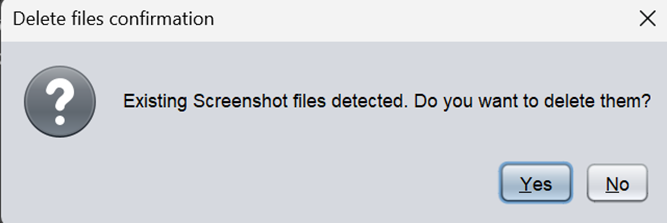


Step 3.1: On selecting Start Button



A window is prompted where the user can choose / create folder / path to save the screenshots in their local storage.

If the user does not choose the path the screenshots will be saved in default folder.



A popup will appear if users choose the folder where screenshots already existed.

Then, based on the users' requirements, they can choose to delete/ retain the screenshots in the folder.

Step 3.2: If user choose “User Demand” mode,

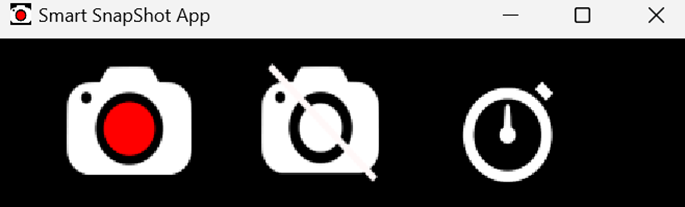


A new window will appear which has a button on clicking this button the screenshots will be taken. This window is draggable so that users can place the window based on the requirement.

Step 3.2: If user choose “Time Interval” mode,

With “Time Interval” selected by the user screenshots will be taken and a flash will appear on the screen to indicate the user that the screenshot has been taken.

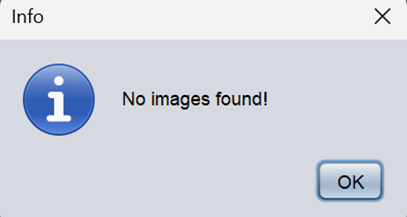
Step 4: Stop Button will be enabled,



“Stop Button” is enabled when the user selects the “Start Button”

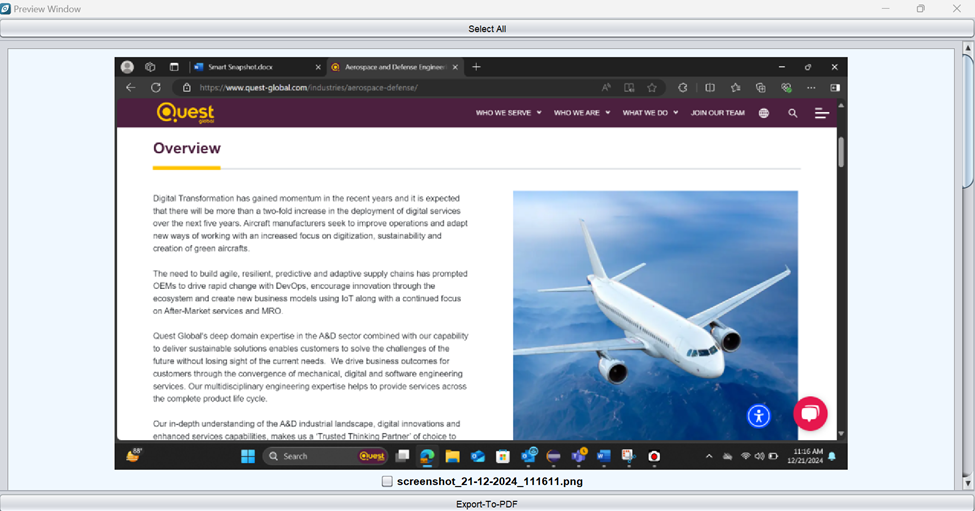
Step 4.1: If no screenshots are taken,

Then on clicking the “Stop Button" button the user will get a notification as,



Step 4.2: If screenshots are taken,

A “Preview Window “is displayed with “Select All” Button and “Export-To-PDF” Button.

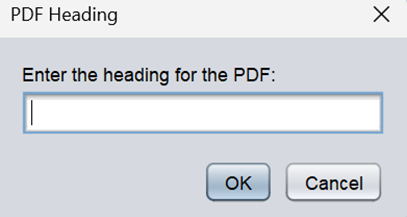
Along with “check box” associated with each screenshot.

The user can select the image by either clicking on the screenshot or on the checkbox associated with it.

The “Select All” button will select all the screenshots in the preview window. And also unselect all the selected screenshots.

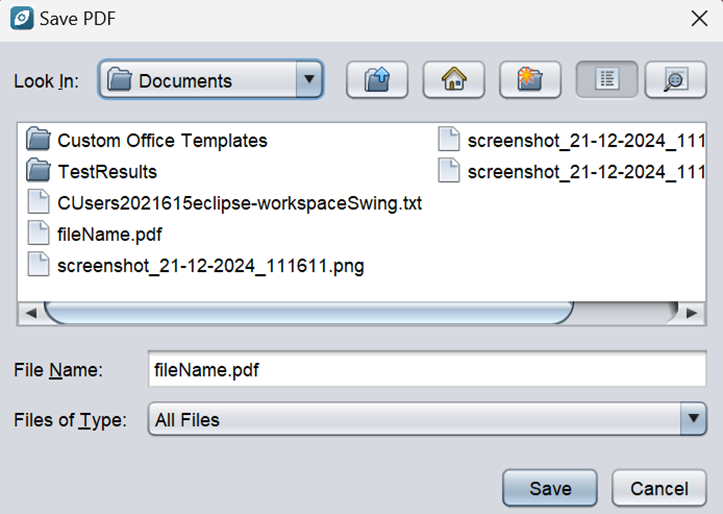
Step 5: “Export-To-PDF” Button,

The user will get a prompt to add the “Heading” to the PDF, if no heading is given by the user, then a default heading will be added “Test Results:”.



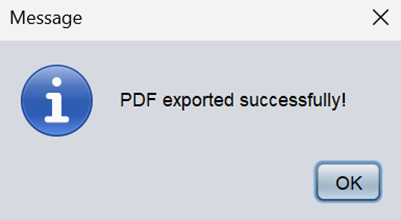
Step 5.1: A new window will appear to save the file,

The user can specify the path or the folder where they want to store the file and also can give the file name by replacing the “fileName.pdf”.



Step 5.2: On saving the PDF,

A popup message will appear to the user,



Step 5.3: On selecting OK,

The saved PDF will appear automatically in the default browser of the user,

