Software Engineering

Screen Share - Spec Document

Harsh Dubey(111901053), Team Member

Overview

In this subject, we are trying to make a lab monitoring system that has the following modules:

1. UX
2. Dashboard
3. Screen Sharing
4. White Board Sharing
5. Content
6. Network

We are implementing the screen sharing feature, enabling students (clients) to share their screen with the professor (server), which includes different functionalities and specifications one must follow to satisfy various constraints.

The screen sharing feature is one the most important features of the project as it ensures fairness during any lab/test session among students and facilitates the professor to take the session smoothly.

Objectives

The screenshare component has 3 parts:

- **View** - To display the screen with various buttons, etc. on both client and server side.
- **View-Model** - To handle various interactions in the UI/View Component using functions and logic.
- **Model** - To capture the image from the client, process it and send it to the server.

I will be handling the View component of the Screenshare module.
The objectives of which are as follows:

Client :

1. There will be a start button on the client screen. On pressing that the screen of that client will be shared to the server.
2. The client will be notified that their screen is being shared by a popup.
3. There will be a stop button pressing which will stop the screen share.

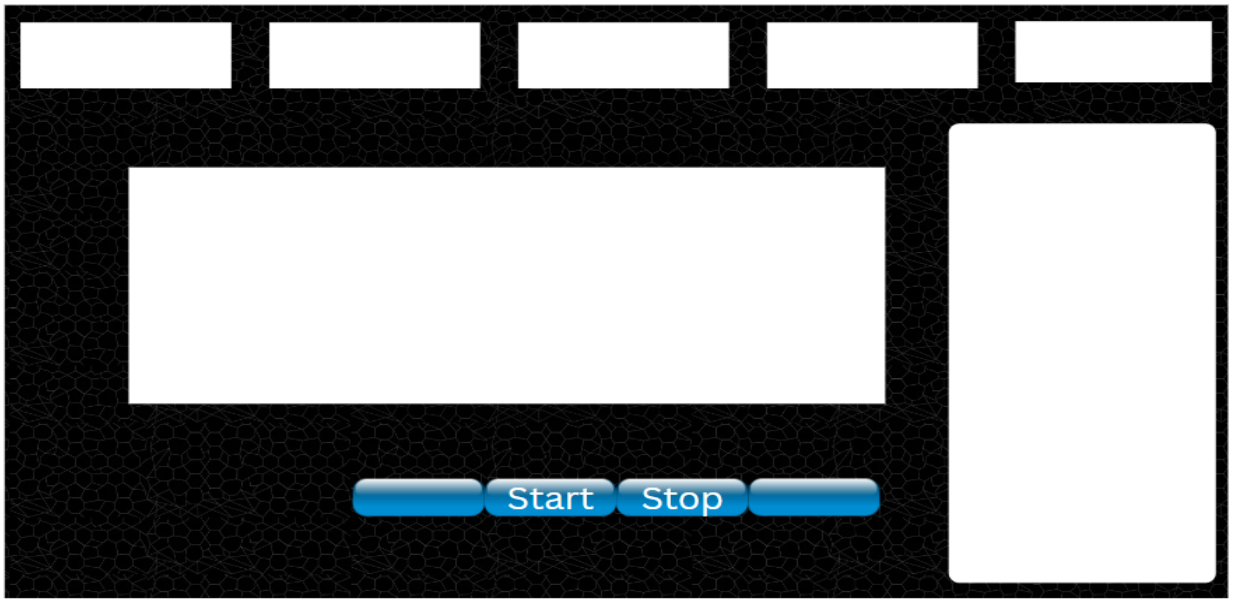
Server:

4. The server can see the shared screens in the screenshare tab.
5. At a time, there will be 9 tiles shared at max on one page.
6. Other tiles will be shown through pagination (having multiple pages).
7. There will be 2 buttons on each side of the screen to go on the next page or the previous page.
8. If there are less than 9 tiles on the end screen, then they will be resized to cover the full screen share area.
9. There will be no next page button on the last page and no previous page button on the first page.
10. Hovering onto a tile will show the pin button , clicking onto which will pin that tile, i.e.,
    That single tile will be in the beginning and shown in full screen, after which other tiles will be shown.

VIEWS

Client Side:

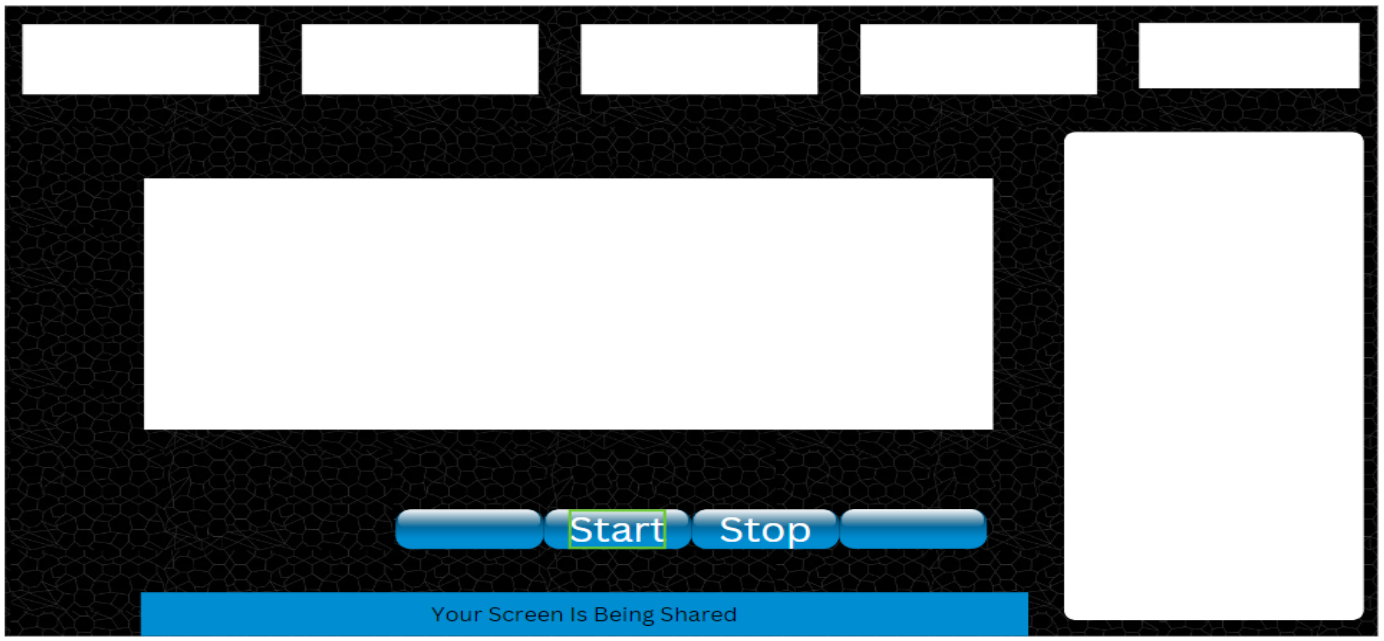
1. **Before Sharing**



Client View (start and stop buttons)

The client screen will look something like this, with a speaker window in between, a chat window on the side and other tabs on the top. This screen will have some buttons at the bottom among which there will be 2 buttons, Start ScreenShare and Stop ScreenShare. On clicking the Start button, it will call the function SendButton_Click() of the class ScreenShareClientPage, which will then call the ViewModel and start sharing the screen of the client.

2. **During Sharing :**

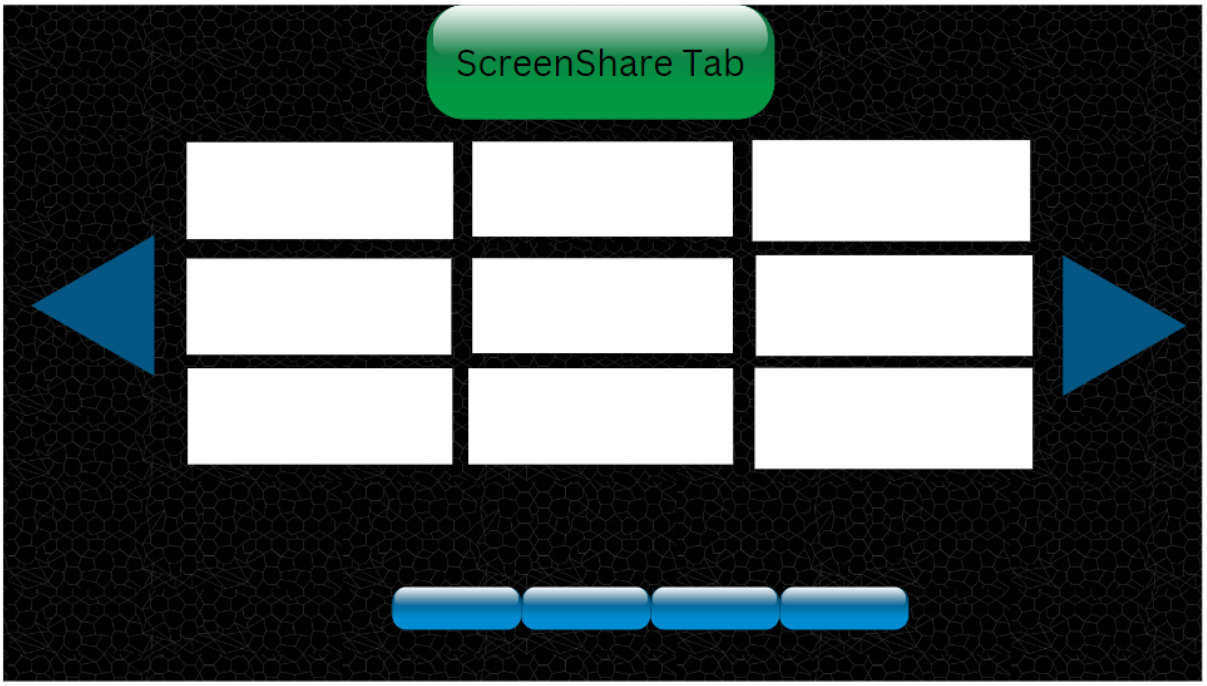


Client View (Screen is being shared)

There will be a popup continuously shown while screen sharing is happening, telling the client that currently their screen is being shared to the server.

When the screen is being shared, clicking on the Stop ScreenShare button will call the function StopButton_Click() of the class ScreenShareClientPage, which will call the ViewModel and it will stop the screensharing.

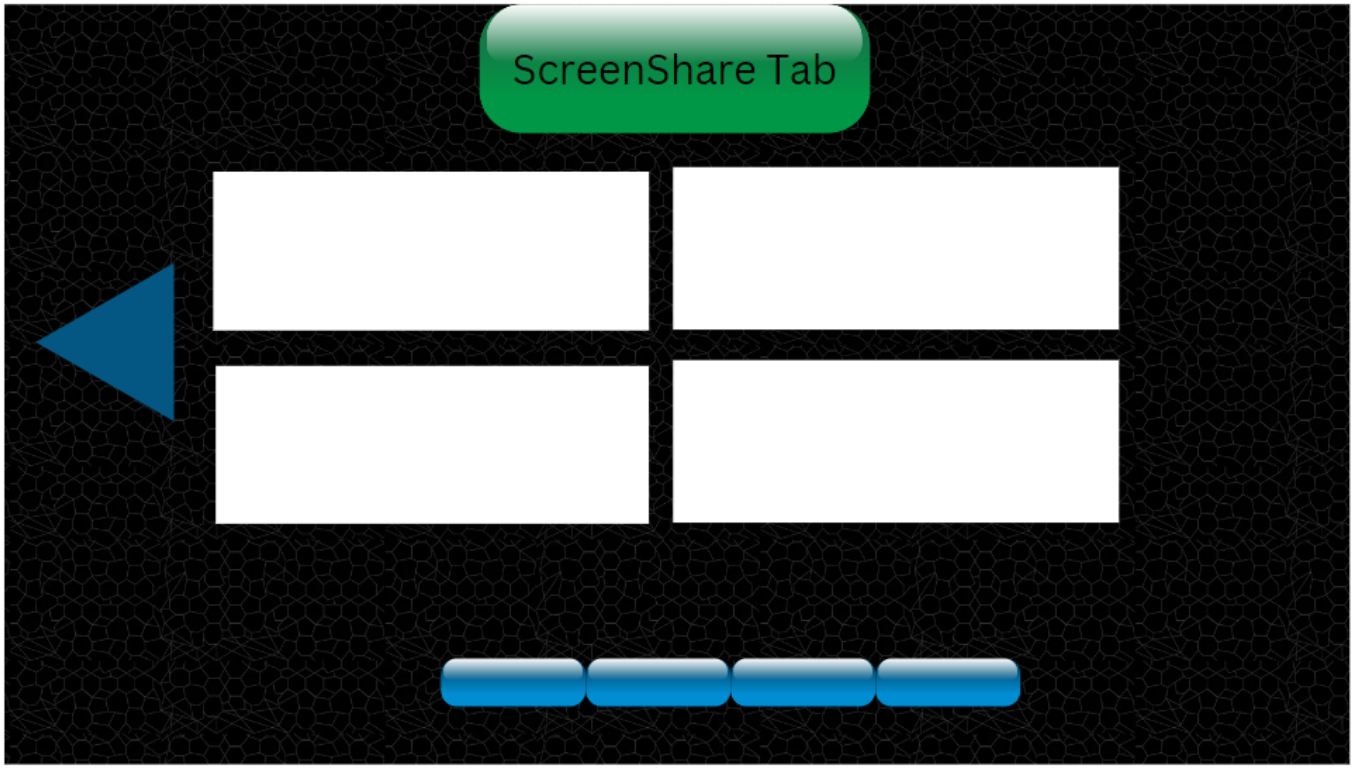
Server Side:



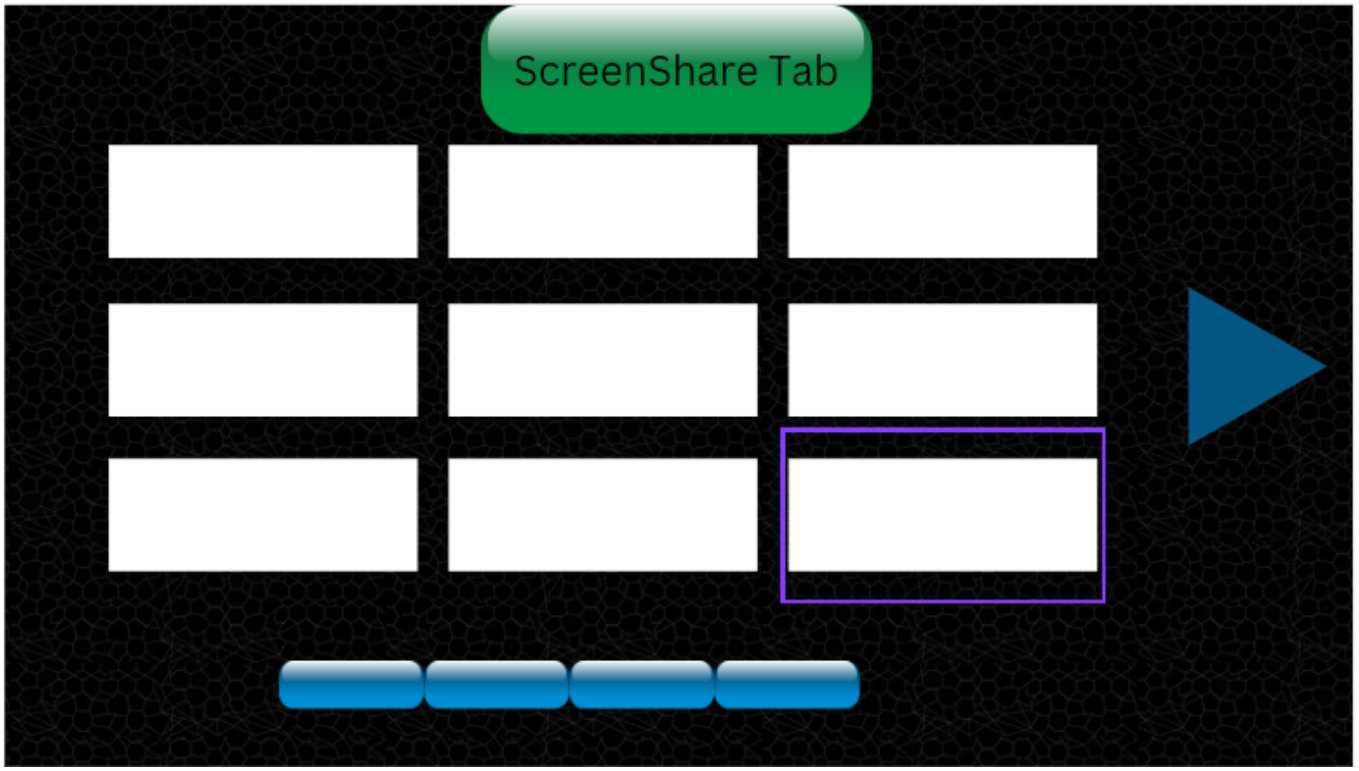
Server View (In a middle page with all 9 tiles)

When the server side moves to the screenshare tab, they can see the screens being shared in the form of tiles. The default page which opens at first will be page 1. Thereafter, on whatever page the server left at last, that one will open when moved to screenshare tab again. On one page there will be at max 9 tiles shown, in the form of 3x3.
The server will have some buttons on the bottom of the screen and also 2 more buttons -> One on the left side (previous page button) and one on the right side (next page button). Clicking on the next/previous page button will call the function PageChangeButton_Click() of the class ScreenShareServerPage. This function will call the ViewModel with input as the new page number, the ViewModel will give the new page which the View will show.

When the server is on the last page and the number of tiles is less than 9, the remaining tiles will be resized to cover the space allocated for the full 3x3 tiles and the next page button will be hidden. Similarly when on the first page, the previous page button will be hidden.



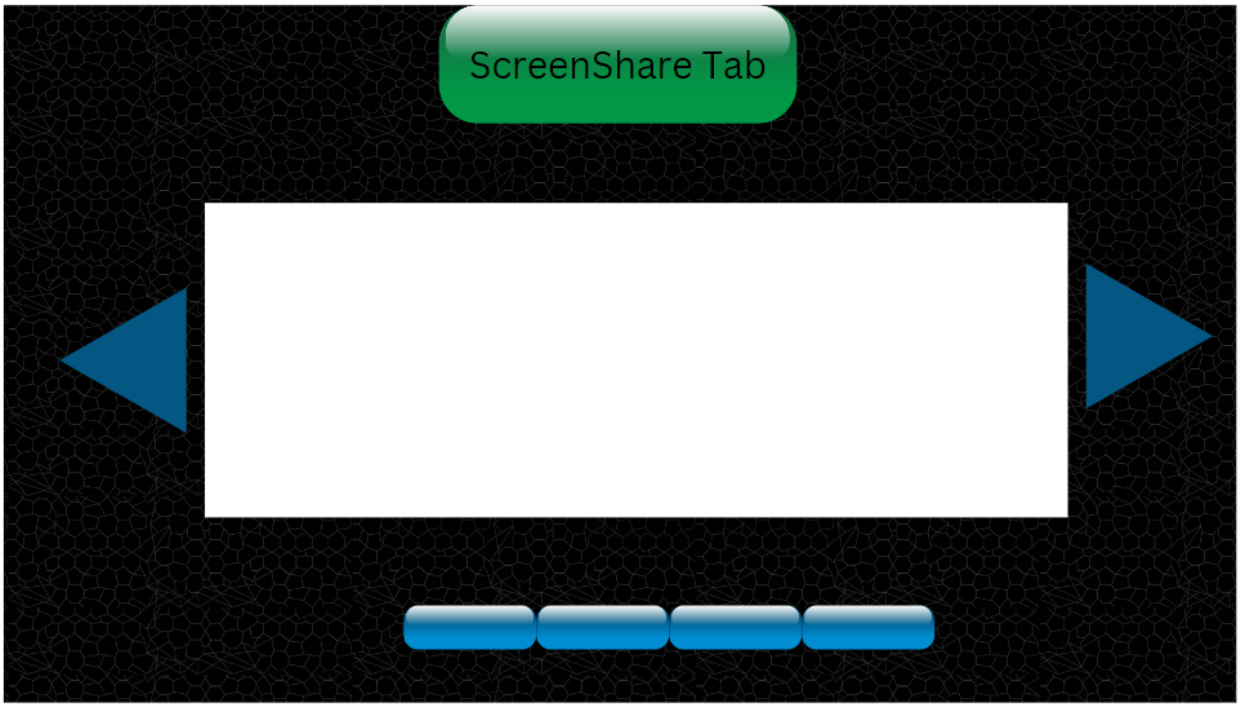
Server View (On the last page)



Server View (On the first page)

Hovering on all of these buttons will display their names. When hovering over a tile, the server will be able to see a “pin” option. Clicking on the pin option will make the selected tile large enough to cover the entire screen and this will be put on page 1.

All the pinned tiles are moved towards the beginning according to their page numbers, and only after that the 3x3 tile view will be displayed. The pin button when clicked will call the function PinButton_Click() of the class ScreenShareServerPage (with the input as IP of the selected user to pin) which will call the ViewModel and the selected tile will be pinned.



Server View (Pinned Tile)

It will look something like this, where the pinned tile is displayed in max size and pagination is still available.

Classes

ScreenShareClientPage //Class handling the client side view

Constructor()
// Instantiate ScreenShareClientViewModel Object
public void SendButton_Click()
//Activates when the send button is clicked on the UI
public void StopButton_Click()
//Activates when the stop button is clicked on the UI

ScreenShareServerPage //Class handling the server side view

Constructor()
//Instantiate ScreenShareServerViewModel Object
public void PageChangeButton_Click(new_page)
//Activates when either next page button or previous page button is clicked on the UI
public void PinButton_Click(IP)
//Activates when Pin button is clicked for a specific tile

Summary and Conclusions

This document tells about the ScreenSharing component of the PlexShare app being developed for the Software Engineering Course. The Screensharing component has 3 parts, namely ->
View, ViewModel and Model. This document covers the details about the View component which is going to be implemented. The View has 2 classes, ScreenShareClientPage and ScreenShareServerPage, which deal with the view of the client and the server respectively. Both classes have their own methods which handle the interactions between the user and the UI and the VIEW sends data to the ViewModel to apply the functionality for those interactions(mainly buttons). This doc also covers what is the intended way in which the ScreenSharing component will show the screen to both the client and the server and some details about those.

Future Work

1. Admin should be able to share the screen with all the clients in the teaching mode
2. Allow client to share a particular application (a window, tab, or full screen), allowed by admin if not in test mode
3. Allow sharing screens at different resolution
4. Allow multiple admin support