

Electris Screen Recorder Prototype System

Group Number: **Team 1**

Project Manager: Alec Mabhiza Chirawu

Student number: 2017060056

Task Assignment

Neloy Mahadi Sajjad	2017002106	Architecture design,coding
Tendayi Elishama	2017050043	Requirement specification,coding
Alec Mabhiza Chirawu	2017060056	Documentation,coding
Sailas Nganga	2017060003	System analysis,Testing
Shiham Md Tayef	2017002102	System Testing,requirement specification

Acknowledgement

Firstly, we would like to express our sincere gratitude to our Professor:

Mr Dandelion, for the continuous support of our project practice related research, for his patience, motivation, and immense knowledge. His guidance helped us in all the time of research and writing of this documentation. We could not have imagined having a better adviser and mentor for our studies.

Table of Contents

Introduction.....	5
Glossary.....	6
User requirements definition.....	7
System architecture.....	8
Microservices architecture.....	8
Testing and bugging tools.....	8
System requirements specification.....	9
Models.....	10
System context models.....	10
Use case diagrams.....	11
Sequence diagrams.....	12
State diagrams.....	13
Class diagrams.....	14
User interface.....	15
Clicking the Source button.....	16
Implementation.....	21
Testing.....	25
Functionality Testing.....	25
Usability testing.....	25
Interface Testing.....	25
Database Testing.....	25
Compatibility testing.....	26
Screen Compatibility Test:.....	26
Performance Testing.....	26
Security testing.....	26

Introduction

In the modern world where proof is essential , everything you do need to be recorded for proof and for clarity, also people need to record their videos, audios, and their daily use of computer in their daily life. For that we then come together as a group and look at that effect at a closer range and see that they is need for us to create something that is pretty nice, cheap and easy to use for screen recording. We then look at the available screen recording and we then set that we are going to create something new with the same taste and different receipts.

Usually everyone is eager to know the new things in every new system and for our system (Screen recorder) you can select a specific page to record without affect all other activities, We researched and found out that all the screen recording applications only record only the full screen and ours records even a single page and let you use other pages without a problem.

Due to a time studying computers we understood that , our computers uses a module called electrons for which is the main key for the computer to operate as it does in its daily life. We used python language for coding and node js with NPM.

Glossary

- Node.js - it is used to retrieve files from the disk and here we will use it for storing and maintaining files in our local database.
- NPM - node package manager, is used to run javascript environment.
- Cd - this is a command to open a folder on terminal
- I - means install, it is used to install modules on terminal.
- Rendering - this is where we will record our video and its where its is shown.
- JavaScript - its a language used to write programs.
- Electron - it is used to create a life cycle of an application.
- Cli - its for executing operating system functions.
- Maker-deb - its for creating debian packages.
- Maker-rpm - is used to configure AMX system by using step by step approach.
- Maker-squirrel - a user friendly logo application.
- Maker-zip - a tool to overcome worm virus salga.

User requirements definition

We set down and set the functionalities which users are able to use and if they are okay and friendly:

The system must be user friendly

For every system to be known as a good system, it must be user friendly. First meaning it must be written in language which an user will understand and it must have clickable buttons.

The system must be colorful

Of course every user needs a system which is appetizing and for that we made sure that our system meets those conditions. We set all colors as all other programs like red means danger or exit.

Fast and reliability

Our system must be fast on loading and reliable as in operating. Every user loves something fast and my team is ready to meet those conditions.

Compatible for all systems

For this we will make sure that the system itself will be able to be compatible for all windows applications.

System architecture

Microservices architecture

- Due to the fact that we want to create a software which is similar to those already in the field but now its more advanced.
- we choose microservices architecture method since this will allow us to modify our project in future.
- ✧ We are going to use python language to program this application
- ✧ We will use nodejs, javascript and html to create pages and for response.

Our developing environment is as follows:

- Vscode as our code editor

Testing and bugging tools

- Code IDE
- Buddy we are going to use it
- HeadSpin
- NetBeans we are going to use it for testing our desktop app.

Libraries and apps for creating this application:

- Electron-forge
- Maker-zip
- Maker-deb
- Maker-rpm

System requirements specification

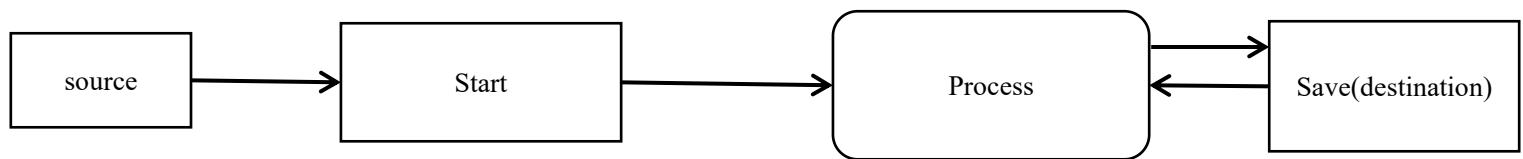
As developers we set and look on the requirements necessary for our system to function well. Our system requirements goes as follows:

- Electron
- Cli
- Maker-deb
- Maker-rpm
- Maker-squirrel
- Maker-zip

Models

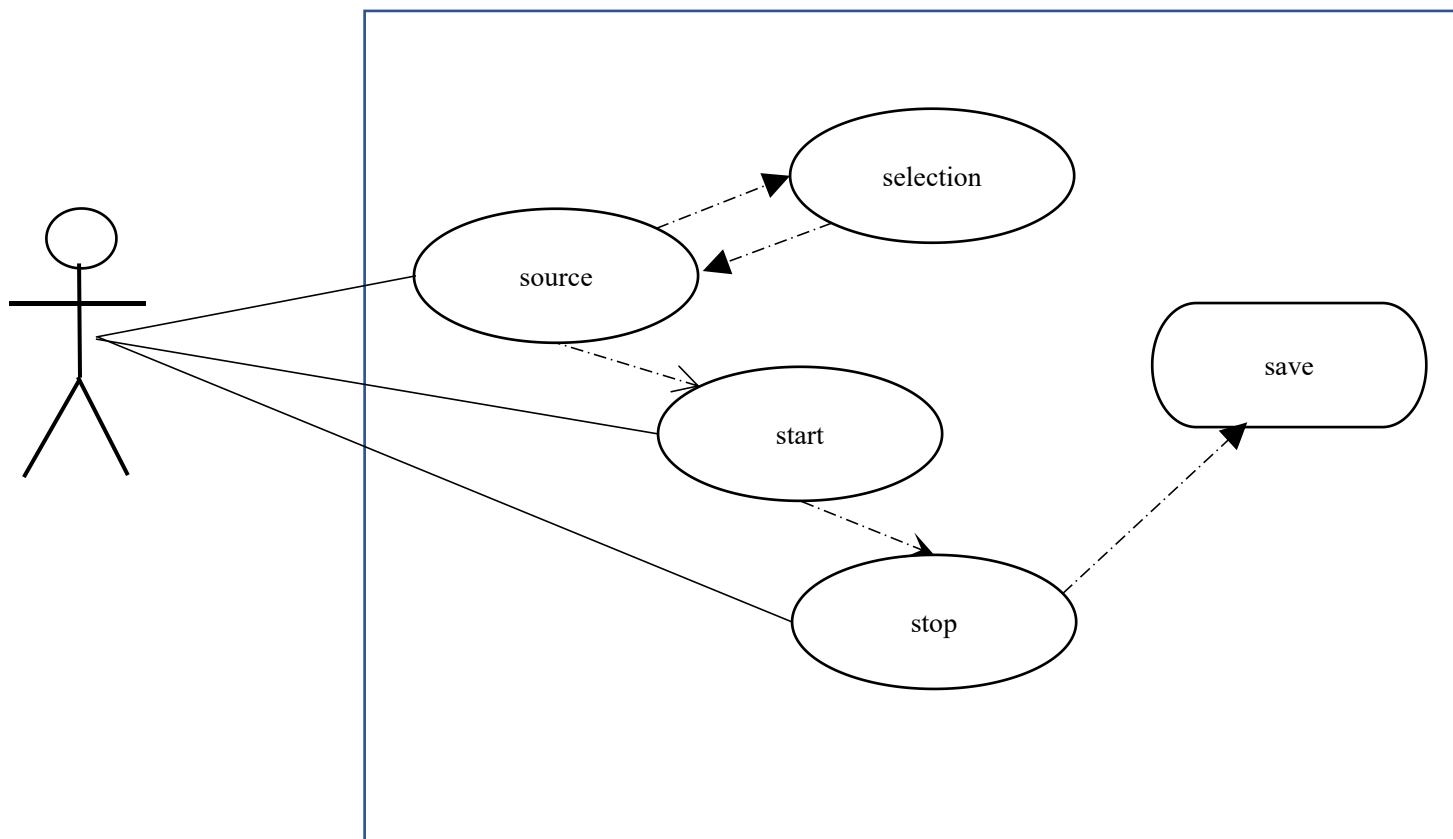
System context models

- Our system environment looks as follows.



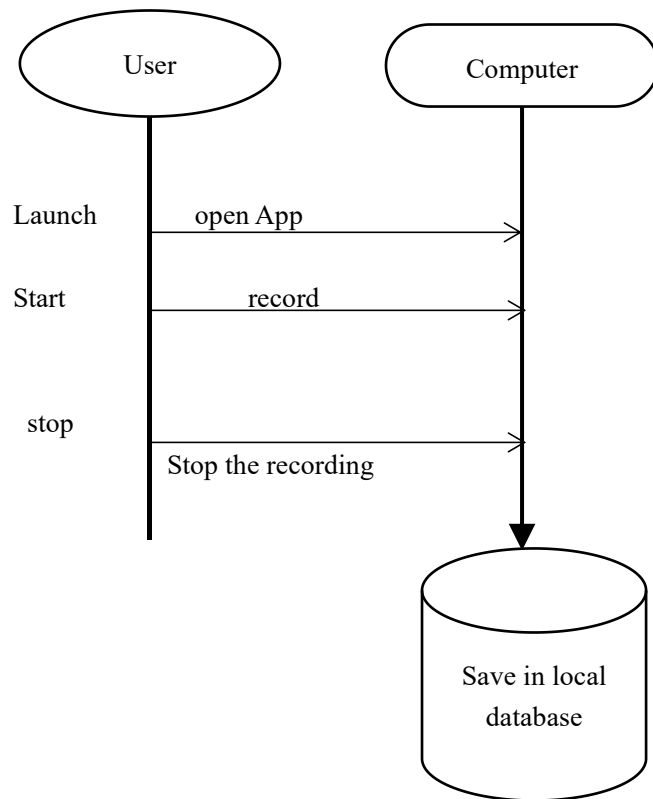
Use case diagrams

- This is how a user is going to access this system with all its stages.



Sequence diagrams

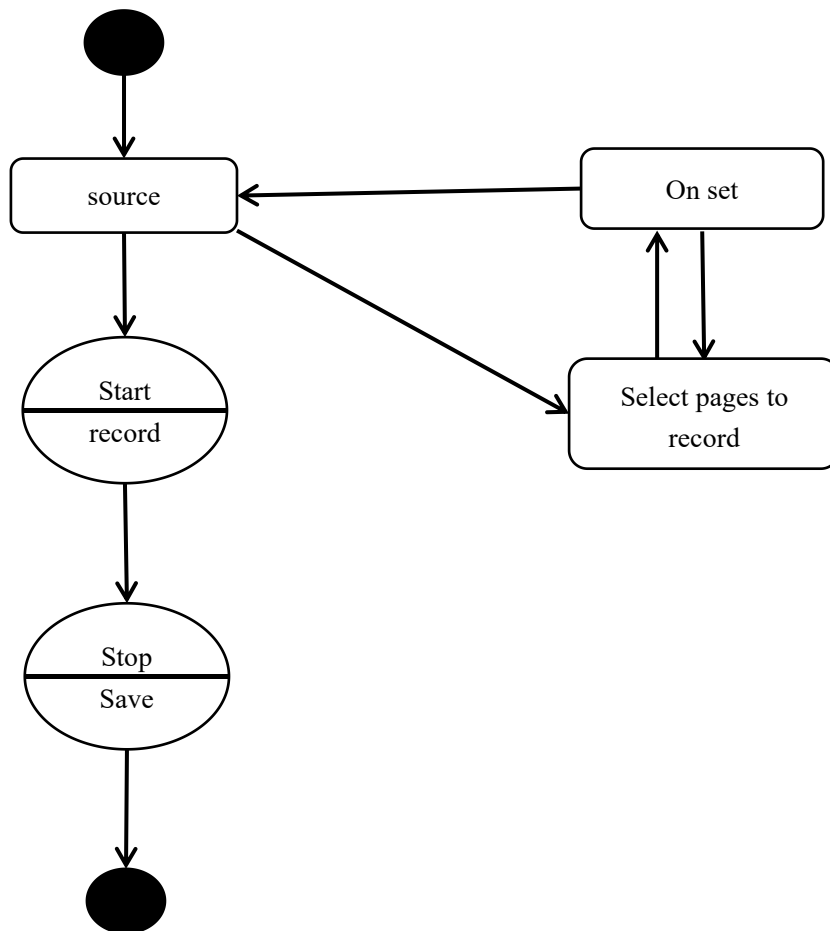
➤ The sequence or the procedure will look as follows:



State diagrams

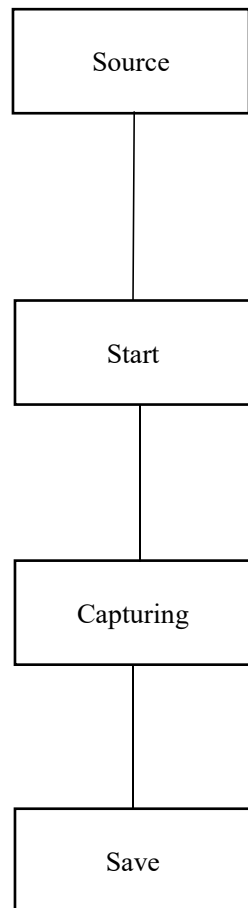
- This is the condition of the system at a given time:

Protocol state diagram



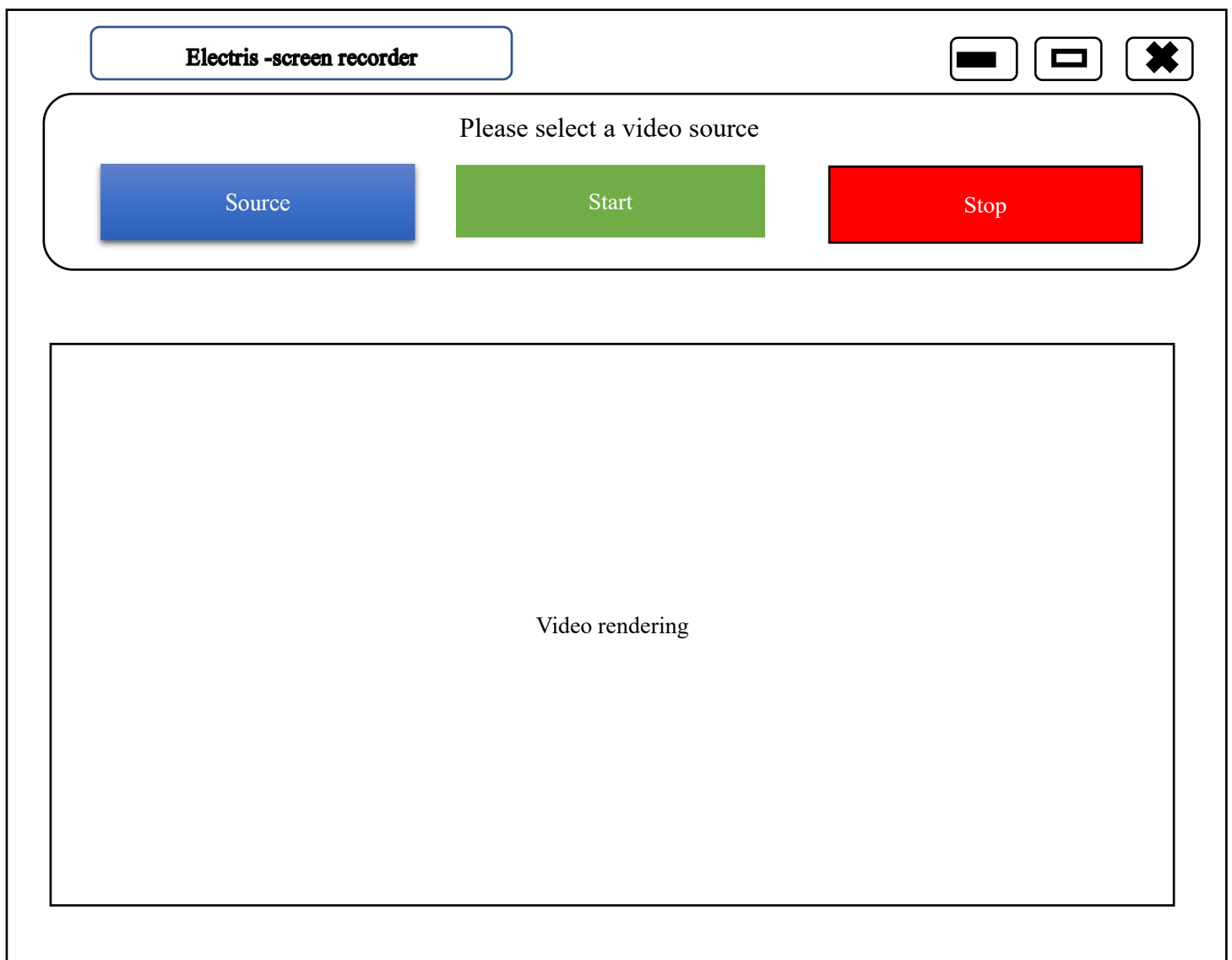
Class diagrams

- We will create this system so that it will be much easy for the users
and all the steps are as follows:



User interface

We designed our interface as follows taking consideration of our user specification and system specifications. To make it easier for the users we put only three buttons on the home page which will extend according to the use.

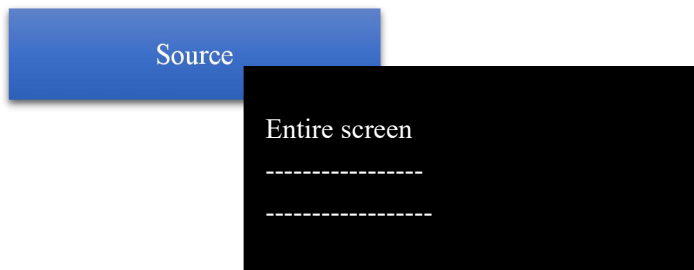


Clicking the Source button

The sequence starts from left to right and we design it like this to make it usable for everyone.

All the open pages which are on the screen will be named here.

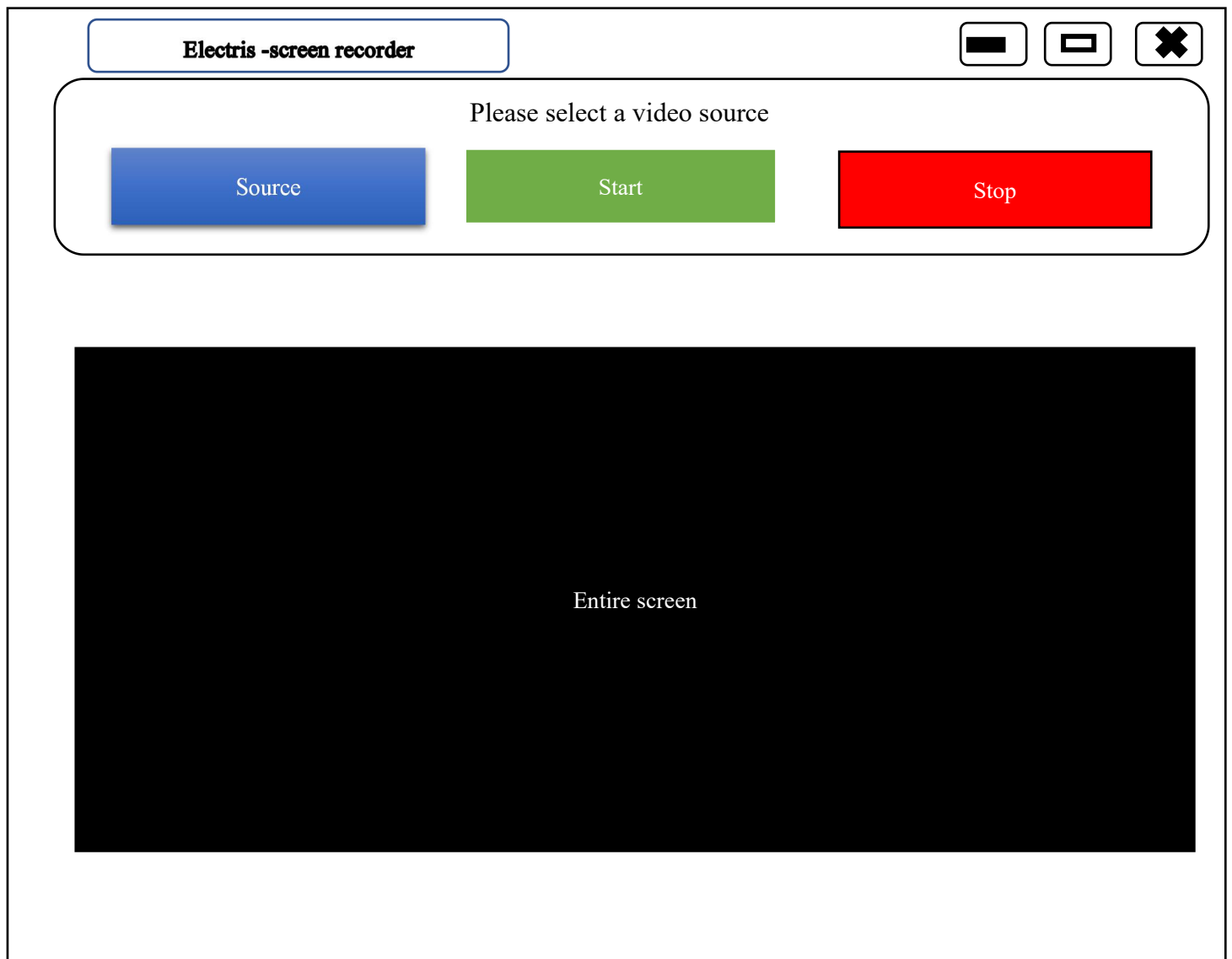
A sample example:



Choosing the one filled on the source list:

After choosing one source file , the list here depends on the number of pages you have opened on the screen and all of them are listed accordingly.

Example entire screen:

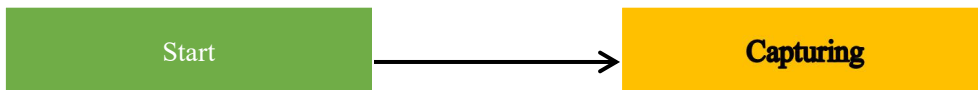


After the source file is on the rendering

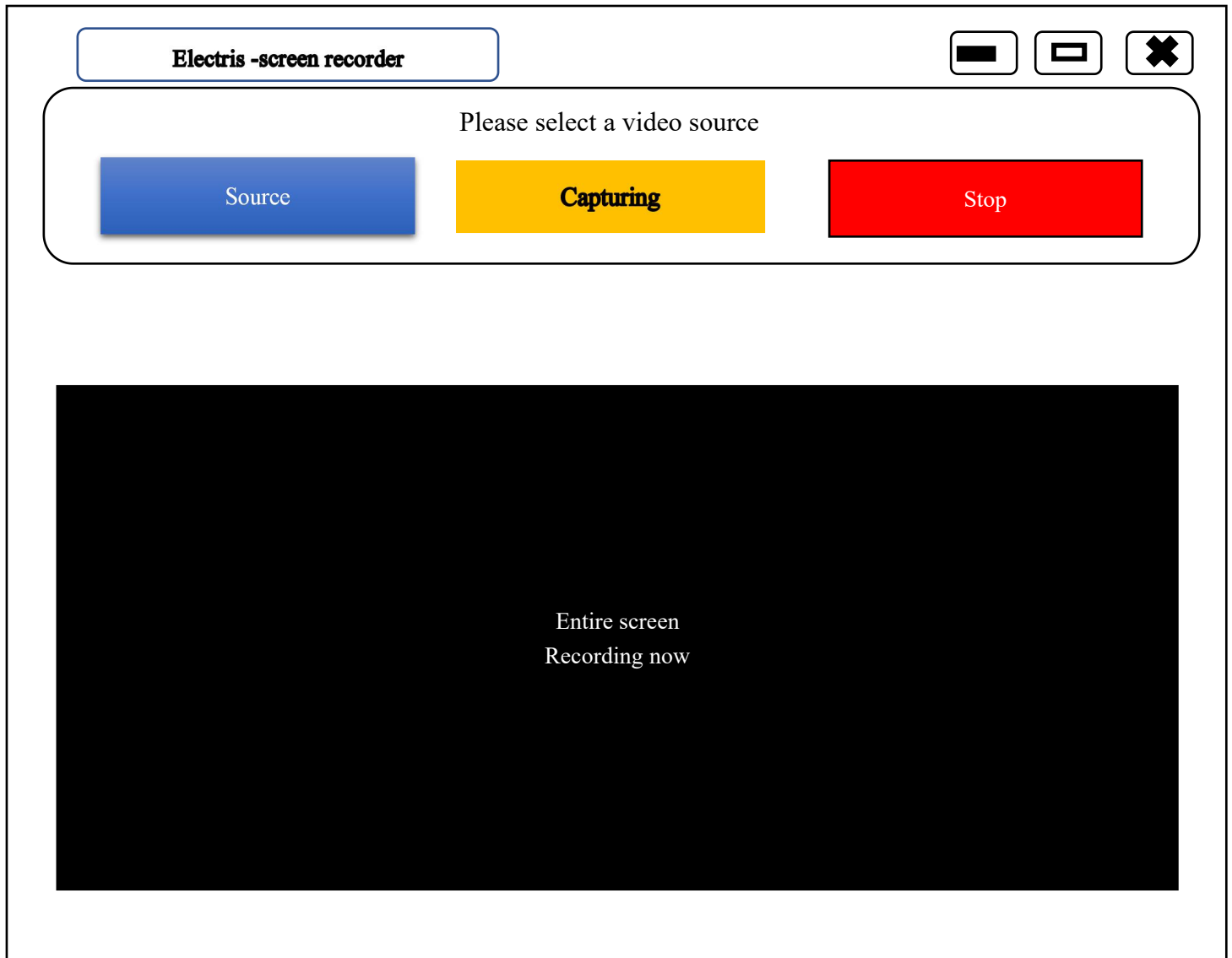
then go and click start:

-For clear and clarity ,after the user clicks an of the source , the user then clicks start meaning we then start to record our screen.

-For the user to know if the screen capture is active , it then changes the start button to Capturing and also changes green color to yellow.



The screen will record and it will be shown on the rendering part , and you can even minimize the screen recording page while recording without an effect.



Clicking the stop button:

- ✧ After the user is done recording , the user will then press Stop on the application and then it will automatically open methods to save the video.
- ✧ That's its it will be done after this stage.



Implementation

➤ We have attached the whole file together with this documentation.

We also create an application and to launch it you only need to click and it will automatically run.

Useful for local host.

Use an text editor .

For vscode

Command to open application in local app

First command : `cd APP`

- The command to install npm modules in your machine for this screen recorder application.

Second command : **`npm i`**

- The command to run the APP

Third command : **`npm start`**

- Incase of errors during installation ,

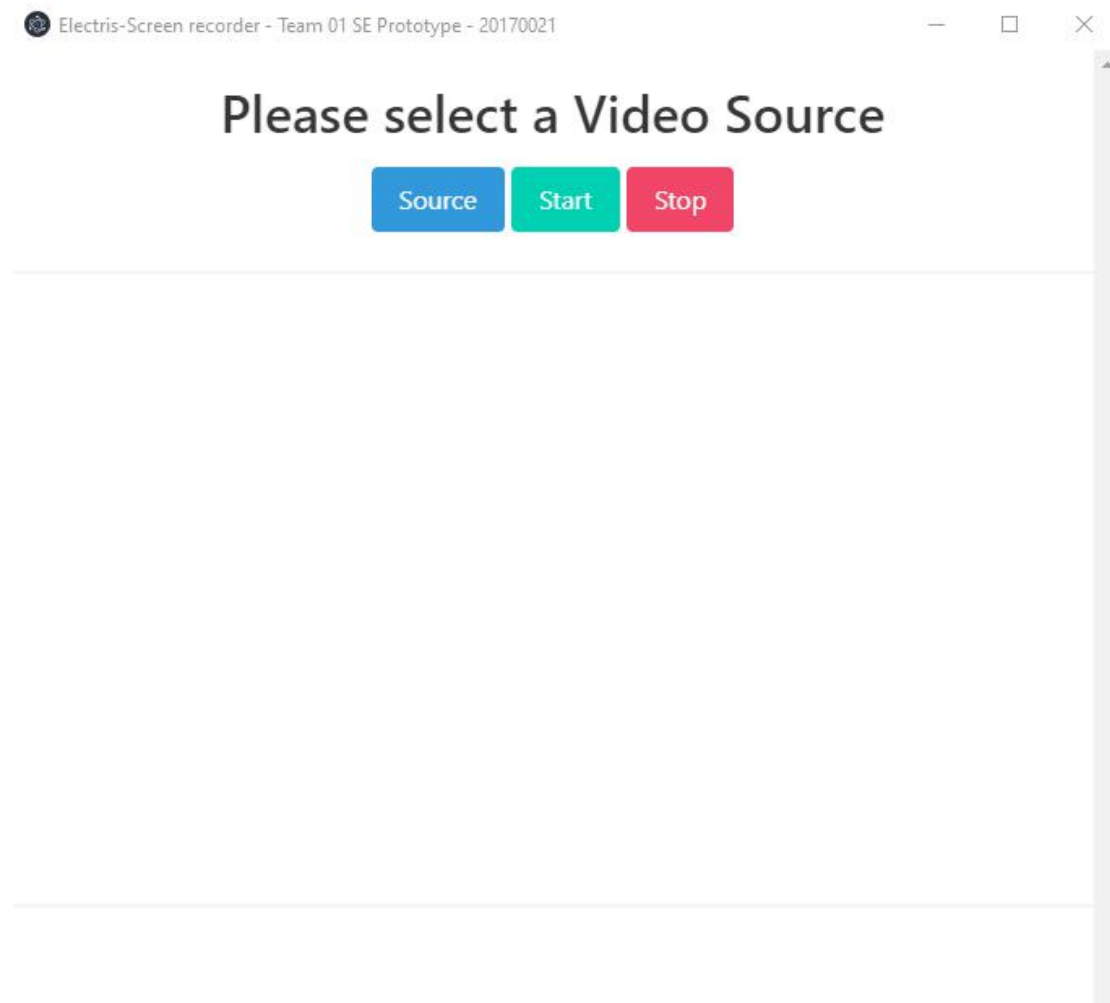
You can use this command : **`npm clean cache`**

- After that start again from second command.

This screen recorder application looks like this:

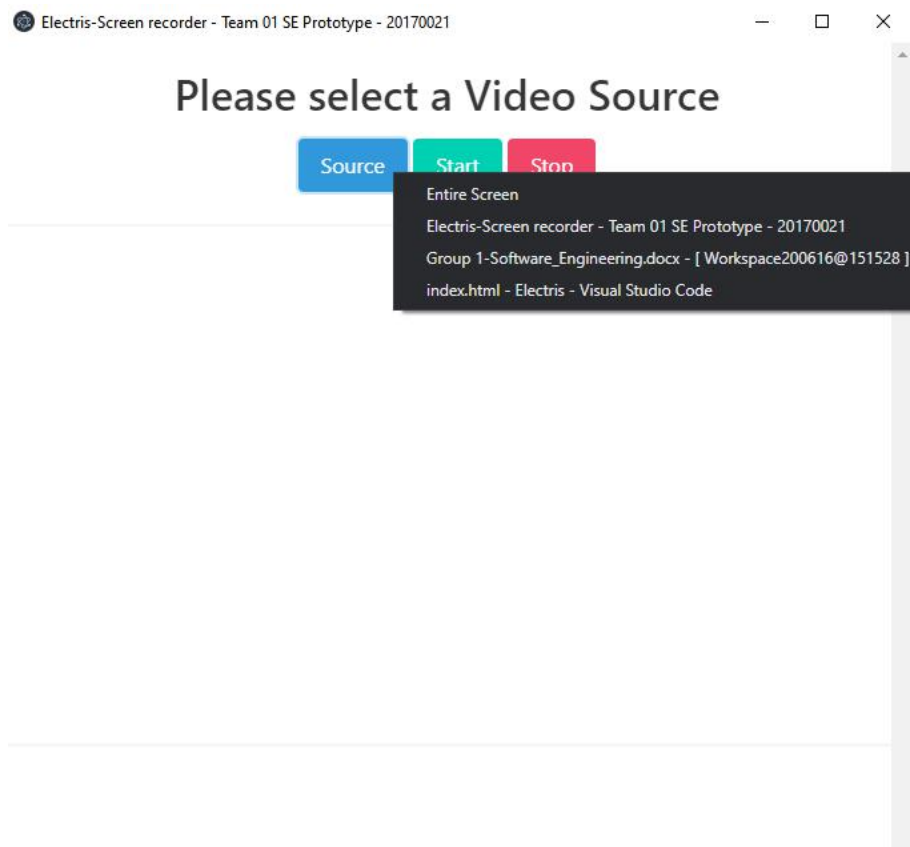
Home page

- The home page have only three clickable buttons which are source, start and stop.



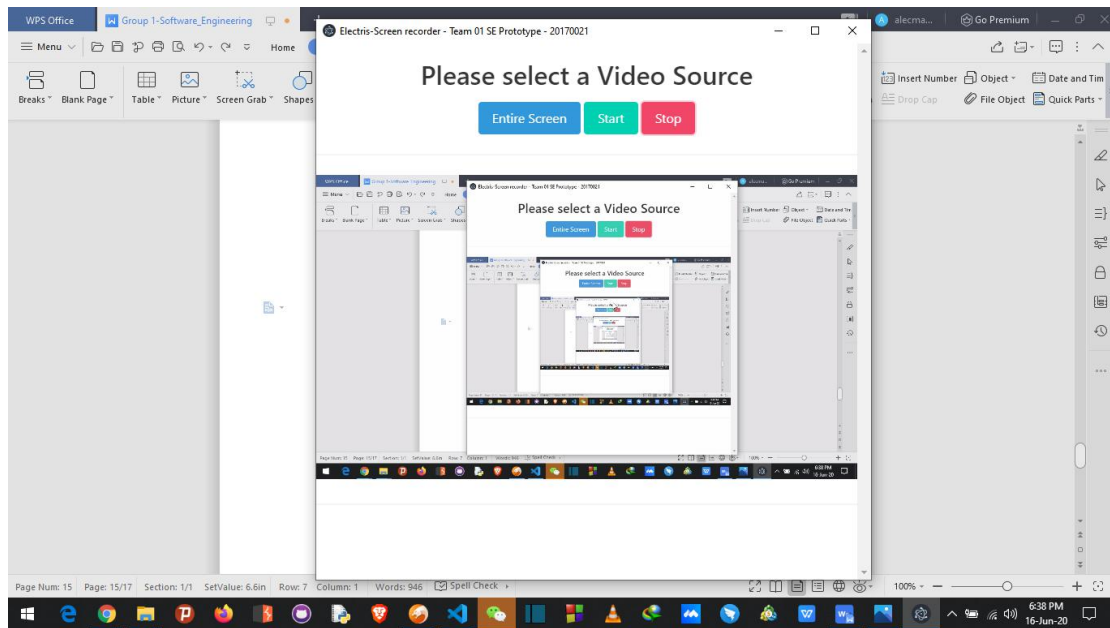
Source page

When clicking the source button , you then need to select the exact page you need to record , from entire screen to recording this screen recorder application.

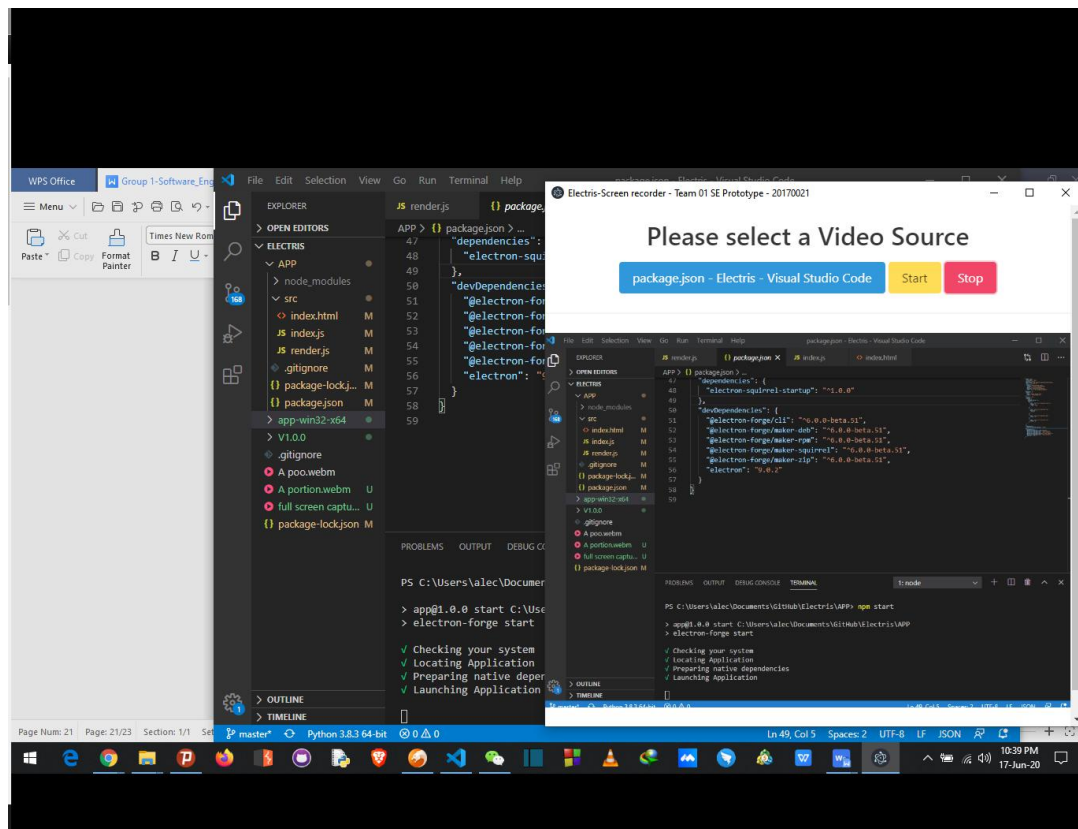


Recording page , (full screen)

The screen will show as follows:



- The most impressive thing we did is that this application can record a portion of the screen without including unwanted parts.



Testing

We have used agile methodology to create this system and for that we we did the following tests.

Functionality Testing

- We have done functionality testing so that we will know if our application is functional.
- Since its a built in App which doesn't need web accessing we tested if it is not conflicting with other apps and everything is fine.

Usability testing

- Usability Testing has now become a vital part of any project. It can be carried out by testers like paid users or a small focus group similar to the target audience of the application.
- We have tested all the navigation buttons and menus and they are easily visible and consistent on all system.
- We also checked on grammar and make sure that everything goes accordingly.

Interface Testing

- We have tested the application, the interface suits all operating systems and they is no app screen break down.
- We also tested if the app is able to support full screen and we as a team makes sure that it is functioning well.

Database Testing

- For database we are using local database so we didn't have anything to test since every machine have an inbuilt database.

Compatibility testing

- Compatibility tests ensures that the application displays correctly across different devices.
- ◆ This would include-

Screen Compatibility Test:

- ✓ Same application in different screens will display differently.
- ✓ It is needed to test if the application is being displayed correctly across screens, operating system, AJAX and authentication is working fine.
- ✓ The rendering of application elements like buttons, text fields etc. changes with change in Operating System.
- ✓ Make sure the application works fine for various combination of Operating systems such as Windows 7, Windows 8, Windows 10 and Windows xp.

Performance Testing

- This will ensure the application works under all loads. Software Testing activities will include but not limited to :
 - ✧ application response times at different connection speeds
 - ✧ Load test the application to determine its behavior under normal and peak loads.
 - ✧ Stress test the application to determine its break point when pushed to beyond normal loads at peak time.
 - ✧ Test if a crash occurs due to peak load, how does the application recover from such an event.
 - ✧ Make sure optimization techniques like gzip compression, time-frame and internal storage enabled to reduce load times.

Security testing

- For this screen recording application , screen recording isn't very important since the security is determined by the computer user.