**Milestone: Use Cases**

**Group 5: Lucas Boehm, Craig Olander, Xiang Pan, Lam Vu**

(Lam Vu)

**Use Case:** User Login

**Actors:** User

**Goal:** To be able to enter the program

**Preconditions:** User needs to know his/her username or email

**Trigger:** Take control the program

**Scenario:** 1. Double click on the program, the “user login” box shows. There is an empty text box, beside it labeled “Username/email”. A box labeled “Send Password” is right below the text box.

2. Type in your username or email

3. Click on “Send Password”

4. User login box is closes, a new box “Password” appear. There are an empty text box with the words “Enter your password” beside and a labeled “Enter” box is on the bottom.

5. Go to user’s email and copy the password just sent to. Paste or type the password inside the box.

6. Click “Enter” box

7. Once entered, the main program box shows.

**Exceptions:** If user forgets username and email, the program will ask 1 of the questions which the user set up already with the questions and answer for each question saved in the system.

Completed and added route to our website funtion

(Lucas Boehm)

**Use Case:** Create Canvas

**Actors:** HTML5 Canvas

**Goal:** To allow users to draw lines

**Preconditions:** User is logged in, config file is loaded.

**Trigger:** Users want to connect two computers together in the tropology

**Scenario:**

1. Once the user logs in the program they will see two boxes. One will be a empty drawling space while the other one will be contain pictures of Computers, hubs, etc.
2. The user will be able to drag one of the pictures into the blank canvas and then able to select a drawling tool.
3. Once the picture is dragged to canvas the user will see a drop down box containing information from the config file in which you can choose what computer, hub, or server you wish to link with that picture.
4. One you have two pictures on the canvas the user will be able to pick up the drawling tool and draw a line that connects the two together.

**Exceptions:**

1. There is no config file to load from. The system will ask if there is another location to load the file from, if there is it will be refreshed to the information otherwise the page will close.
2. If the user drags a picture on to the canvas and there is defined “hub” in the config the system will notify the user and delete the image.
3. When the user draws the lines and they don’t have a end point the system will ask you to please connect the lines between two pictures.

In the process of combination with Xiang

(Craig Olander)

**Use Case:** Configuration Page

**Actors:** user

**Goal:** To have a configuration page that works with the code and is easy to update and add certain code to.

**Preconditions:** User can make his configuration settings

**Trigger:** User wants to change some of the configuration settings

**Scenario:** 1. System displays the configuration page and shows the code that is being used as default settings

2. User clicks “Configuration page” to see the settings

3. If the user wants to make his own individual settings it will show a prompt telling the user that they will be saved and will be advised to save the config page before you make any changes

4. User can type in his own code in the configuration to his/her liking

5. User clicks on the save settings

6. User will be asked to restart the program for his configurations to take an affect

7. Once the restart happens, the program is now his own program to play around with and make it work how the user’s wants it to work

**Exceptions:** a. User misspells some of the code on the configuration page, which will lead the configuration page not to work, make some common coding errors

b. User clicks cancel; System displays a confirmation message, asking whether user wants to discard his actions. If so, then user is taken back to the default configuration settings and nothing is changed.

c. The user is prompted if he wants to change the default settings

d. User will have to know which programming language we used to make the configuration page.

Incomplete.

(Xiang Pan)

**Use Case:** Print/Save

**Actors:** User

**Goal:** To be able to save and print user’s work

**Preconditions:** User logs in with given account password and is finished with his/her work

**Trigger:** User wants to save his/her work and be able to print the work

**Scenario:** 1. Web application has a button that the user can click on and gives the user a series of choices

2. When user chooses “Save work”, the application will ask the user to save his/her work in a designated location

3. After the user chooses a location to confirm and save, there should be another choice which gives the user the ability to print his/her work

4. If the user chooses to print, then the application should be able to print the user’s work on a single page of paper

5. If user chooses “cancel”, then the whole application will cancel and will end the current process

6. After the user chooses to save work, the application will then display a confirmation “Application saved” in a display box.

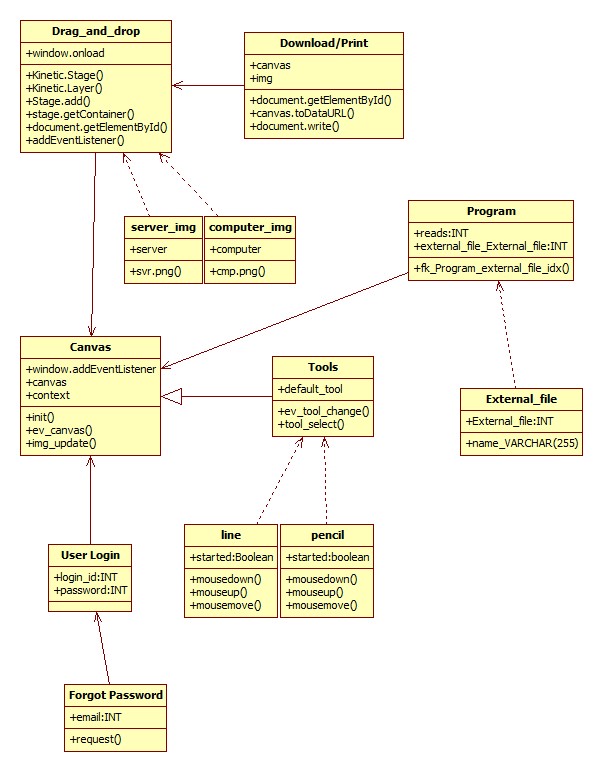
7. User can choose not to save their work, the button “cancel” will cancel the user’s work and exit the application

**Exceptions:** a. User clicks on the “X” on the upper right hand corner, the application will pop up a message box stating that the user must close using the “Close” button.

b. If the user decides to cancel, the current process will be closed and the user will be sent back to the original state without any file being altered

Will be a new canvas using fabric.js

**Milestone: UML**

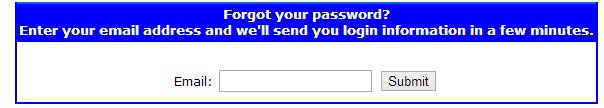


**Milestone: Task List**

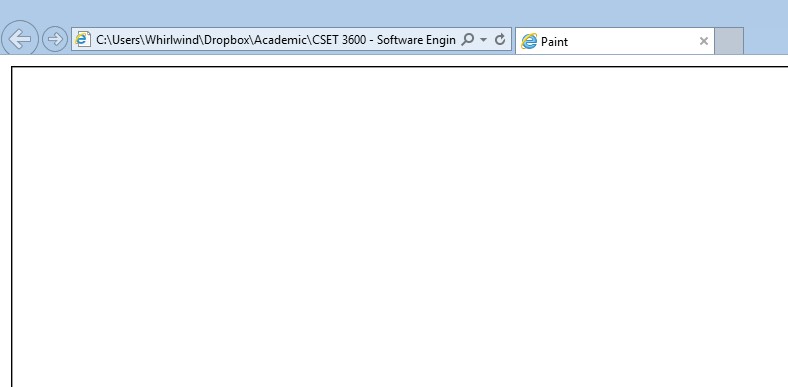
Lam Vu – Design a Startup/Log in page for end user to use. Below is a screenshot of the working program.



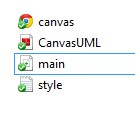
If user forgets the login name or password, the link provided is for the user to submit an email and we’ll provide the user another login information.



This is a still a work in progress. Final form may or may not be changed. As of now, it only works individually. We hope to incorporate this into our whole program in the coming weeks.

**Lucas Boehm** – supplies the canvas. A working canvas is in development. 

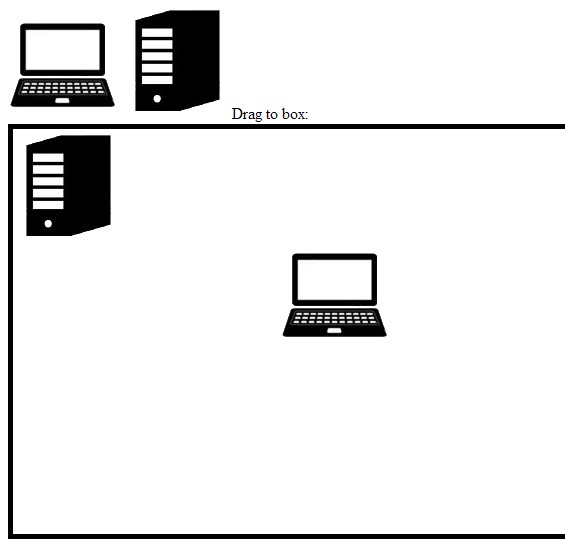
As of now, three files are in place, main Javascript file, Style CSS file, and canvas html file.



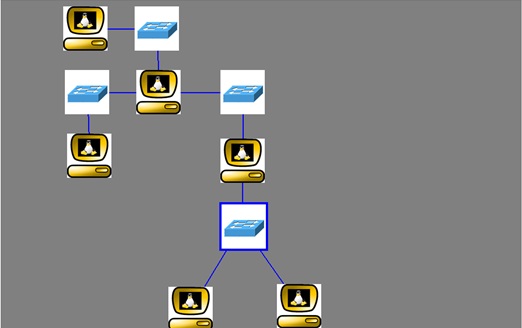
We will build upon our program on the canvas itself.

**Craig Olander** – Read files from network configuration file (network1.cfg) and convert it to objects to use in canvas accordingly. Vm Gemini, vm Nfs, vm Intfw, vm Dmz, vm DefaultGW, etc. will be read and form into objects in the canvas.

**Xiang Pan** – Drag and drop feature for the program. Below is a work-in-progress screenshot.



User will have the ability to drag on an object and drop them on a canvas, server or computers will be objects. We will plan to take the reading file from Craig Olander’s work and name each virtual machine for user to choose. We plan to make a network topology similar to the picture below. Proper connection will be made as well. We also plan to make a print-out of the canvas page after user finishes the program.



Canvas that we have right now:

