**­MineService**

CJ Miller

Collin Trowbridge

Brandon Hennessey

3rd February 2016

Team D

Milestone 4

­

**Introduction**

MineService is a Client-Server application that would help monitor and run Minecraft servers across different systems. It can be split into two different feature sets. One for dealing with physical systems, such as resource allocation, while the other dealing with people, such as support tickets. Between these two sets there will be many sub features.

For physical system features the main ones will be on control. We’ll have ways to start, stop, and restart the server and also automated versions of those. There will also be ways to upload, edit, and delete files on the server.

The people system will include both support and permissions systems. The support side will allow people to both submit and resolve issues in a centralized semi-public place. The permissions system will control who has access to what physical system features and what things they can see on the support side.

**CSSE371 Change Log**

**Milestone 7:**

Added business model canvas

Added value proposition canvas

Reported status of changes based on usability lab

**Milestone 6:**

Added pictures for new completed features.

Updated pictures for some completed pictures.

**Milestone 5**

Updated the high-fidelity prototype images that changed.

Added “Feature Completed” section.

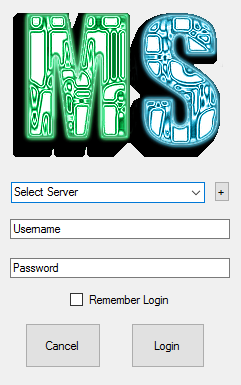
**Milestone 4:**

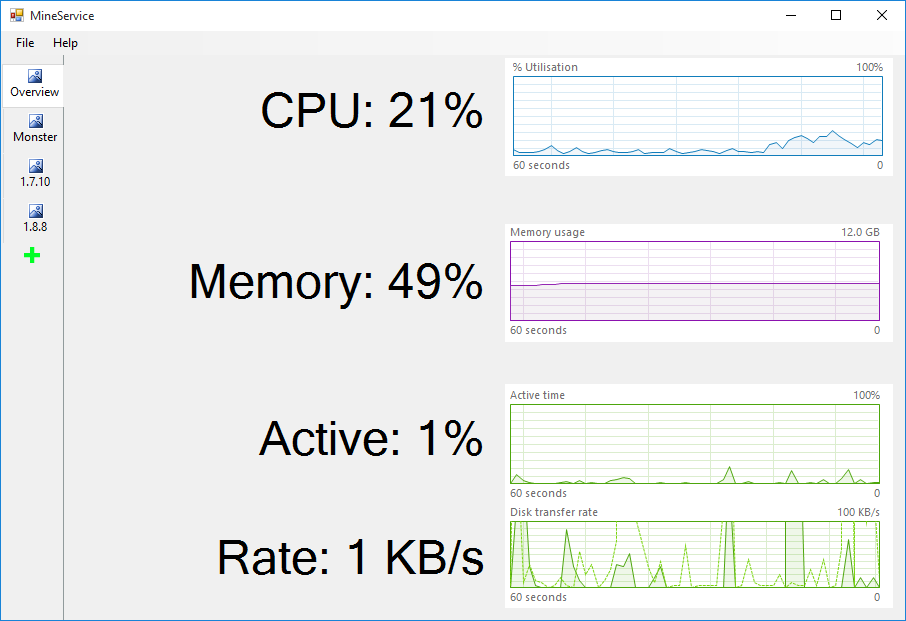
There has been no significant changes since Milestone three. We have implemented the changes our client suggested and still working on making some of the pages more to their liking.

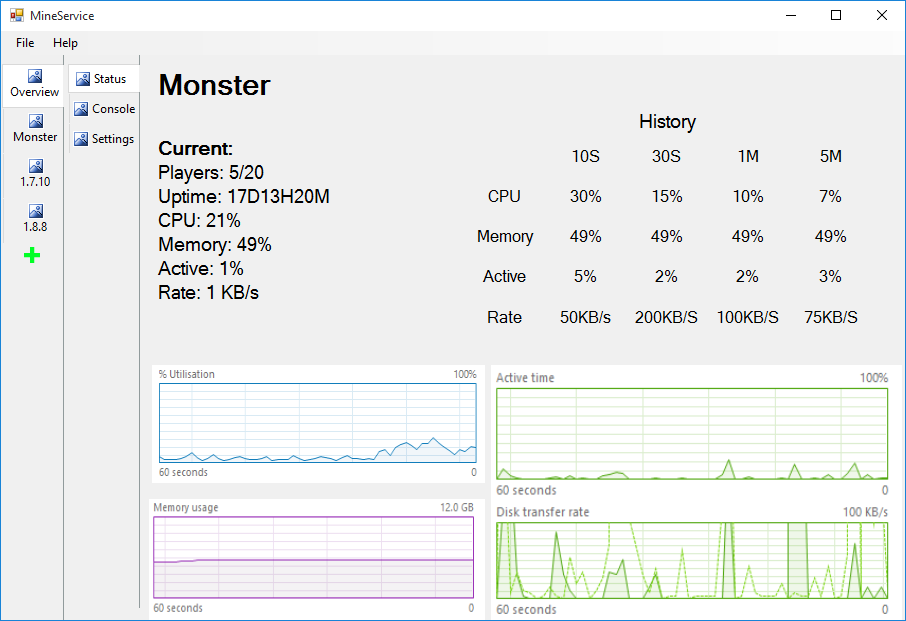
**User Stories**

* **Epic**
  + **Performance**
    - As a developer I want to give typical users better performance when running Minecraft users so that they can have a more enjoyable Minecraft experience.
  + **Time Saving**
    - As a developer I want to give typical users the ability to minimize their time spent on maintenance activities so that they can have their server up and running quicker.
  + **Ease of Use**
    - As a developer I want typical users to be able to easily accomplish their everyday tasks so that they do not have to waste time looking up fixes or syntax.
* **Feature**
  + **Performance**
    - **VM Overhead**
      * As a Server Administrator I want to reduce CPU and Memory Usage so that I can dedicate more to my clients.
      * Condition of Satisfaction:
        + Drastically reduce the number of required VMs per client.
    - **Memory Limit**
      * As a Server Administrator or Minecraft Admin I want to be notified when a Minecraft Server is approaching its memory limit so that I can see if it is an issue or just user spike
      * Condition of Satisfaction:
        + I am notified via text or email within a minute of a condition being met.
    - **Frozen Servers**
      * As a Server Administrator or Minecraft Admin I want to be notified when a Minecraft Server is frozen so that I can fix it promptly
      * Condition of Satisfaction:
        + I am notified via text or email within a minute of a condition being met.
  + **Ease of Use**
    - **Terminal**
      * I’m a person who is responsible for a couple of Minecraft servers but I don’t like to program. I prefer not having to memorize code or commands and type it in the terminal because there are so many of them. To make my life easier, I need to have a User Interface which would do all the stuff the terminal commands do with just a click of a button.
      * Condition of Satisfaction:
        + Acknowledge that I don’t have to use terminal to manage my servers
    - **Multiple Applications Open**
      * I’ve realized that the less number of applications I have open, the less CPU usage and memory leaks. Having multiple applications open can potentially slow down the servers and is also really hard to manage since I need to switch between them at the right time for the right thing. It would be so much better if I just had to have one application open which would display all the information which I seek from multiple applications and perform an action based on that.
      * Condition of satisfaction:
        + Just one User Interface which would display all the information needed
  + **Time Saving**
    - **Terminal Commands have Multiple Steps**
      * I’m an individual who owns a few Minecraft servers, however I don’t write programs as my profession. I would rather not have to enter commands into a terminal, because it takes time to memorize the commands, and if I don’t memorize the procedure of how to navigate the terminal, I will have to take time to write a document with steps outlining the commands I have to execute. It would save me time if I only had to click a button on a graphical user interface, and all of the terminal commands would be done for me.
      * Conditions of satisfaction:
        + Make sure that I don’t have to enter any commands into a terminal to manage my server.
    - **Regularly having to login to Check the Servers**
      * I’ve been told that Minecraft servers with a lot of mods on them can cause memory leaks, and memory leaks can cause a lot of memory, disk, and possibly CPU strain on my server. This can slow down all of my servers, and possibly cause them to crash. Therefore, I must log in to the windows application every three days or so to check the memory, disk, and CPU usage of each server. If a server is taking up too much hardware, I have to hope that no one is on the server, so that I can stop the server, then start it again. If there are people on the server, I either have to stop the server anyway, which will kick off all of the people who are on the server, or I have to wait until another time, log in again, and check and see if anyone is on the server. It would be much more convenient if a server automatically restarted if no players were on and the combination of CPU, memory, and disk usage reached a certain level. Also, if the server was not able to be automatically restarted, due to players being on the server, and CPU, disk, or memory usage got too high, I would like to receive a text message, e-mail, or both, describing the issue to me, so that I can manually restart the server if I saw the need.
      * Conditions of satisfaction:
        + If there are players on a server, and the combination of CPU, memory, and disk usage exceeds a certain level, notify the server owner via their specified means of contact, either text message, e-mail, or both.
        + Have the server automatically restart if the combination of CPU, memory, and disk usage exceds a certain level, and there are no players on the server.

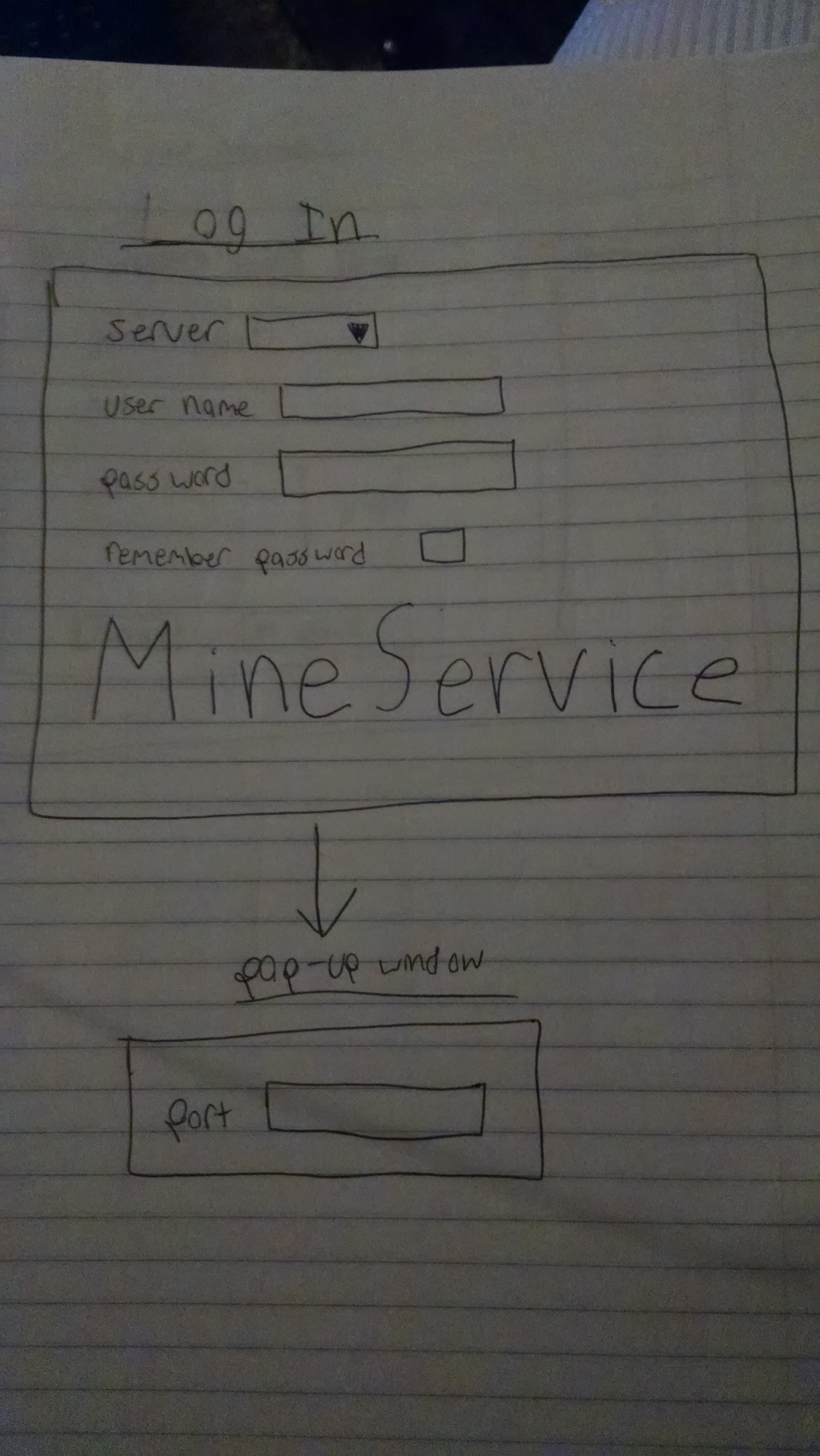
**Low-Fidelity Prototype**

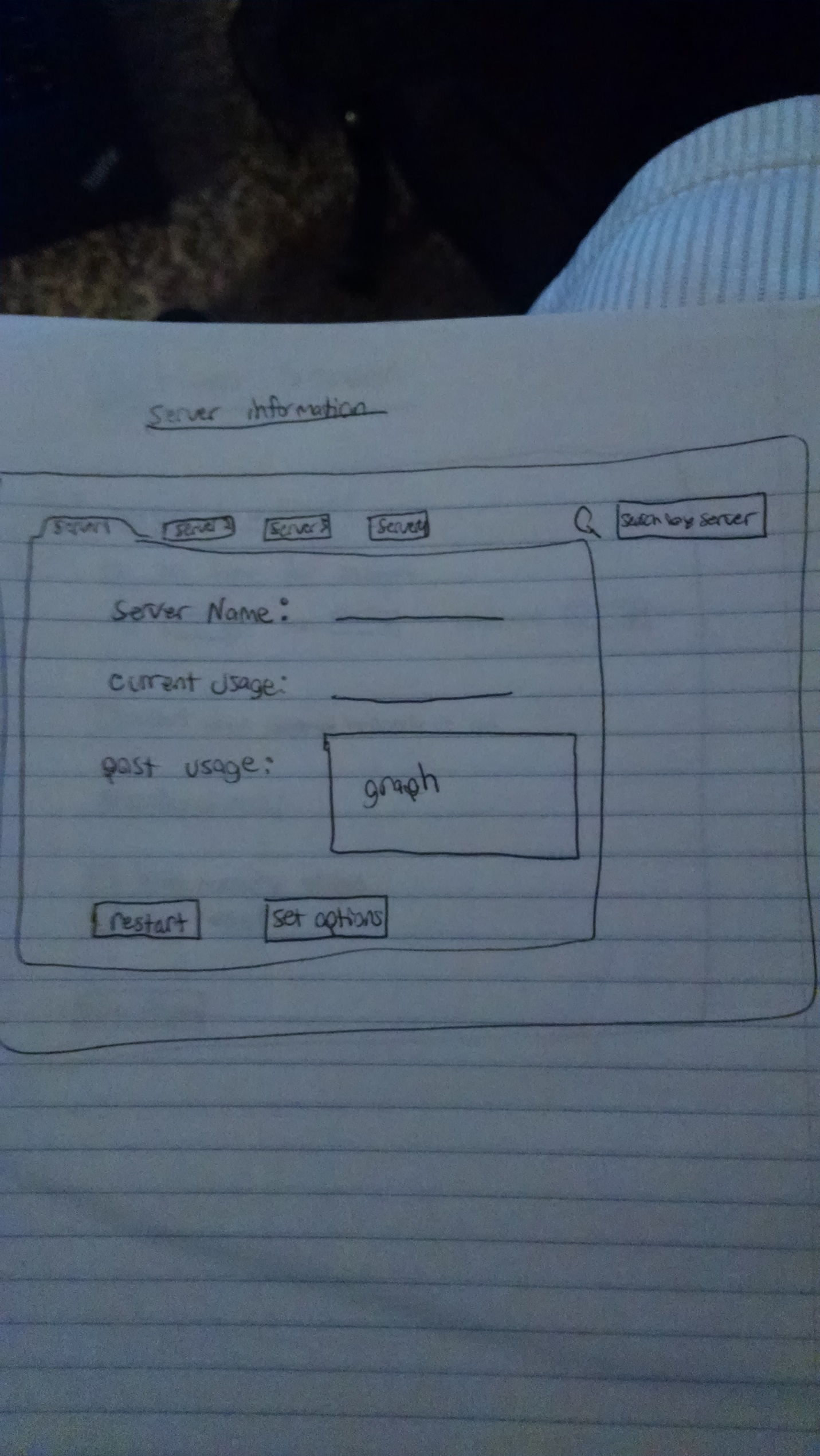


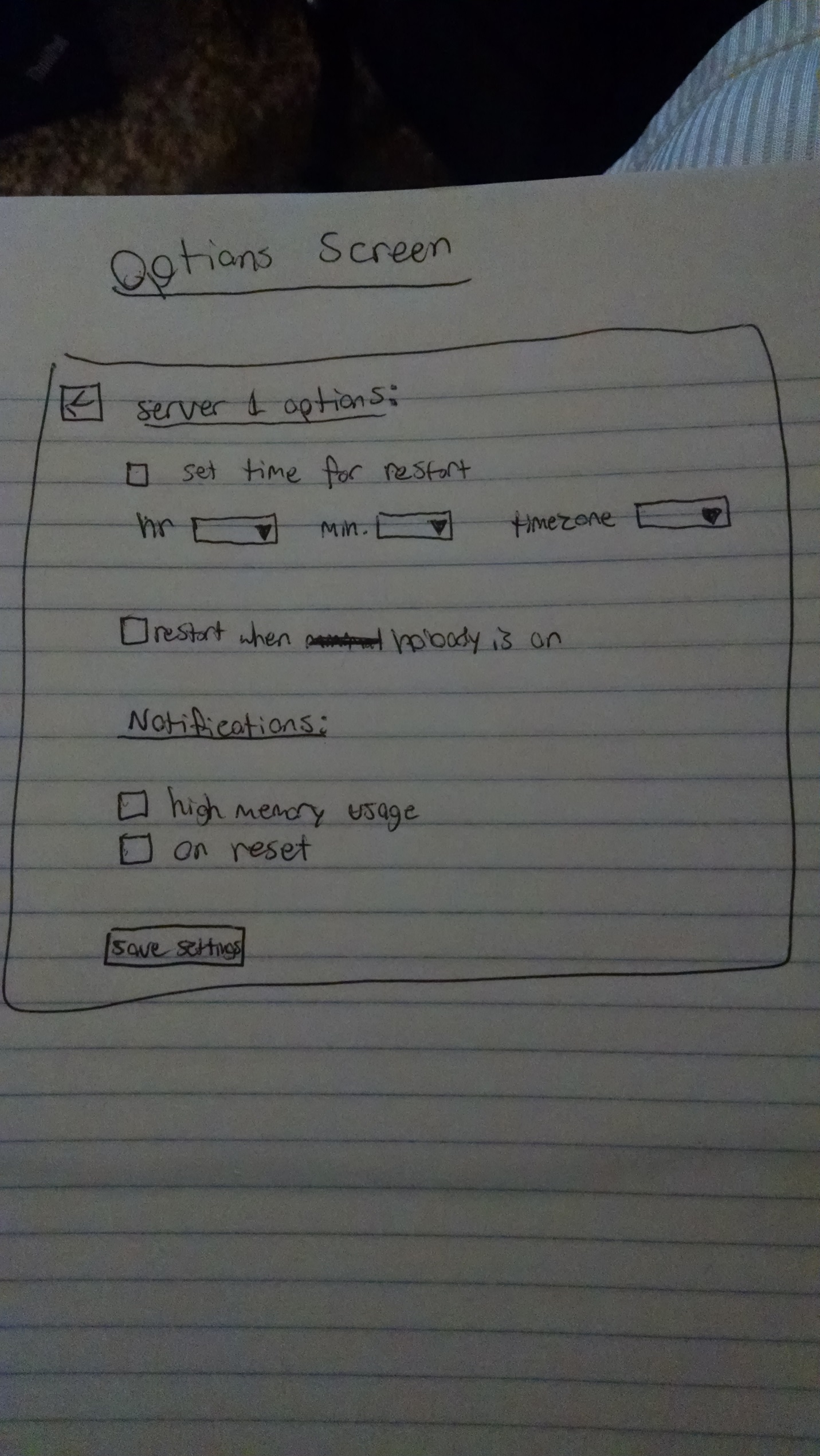




**Alternative Design**







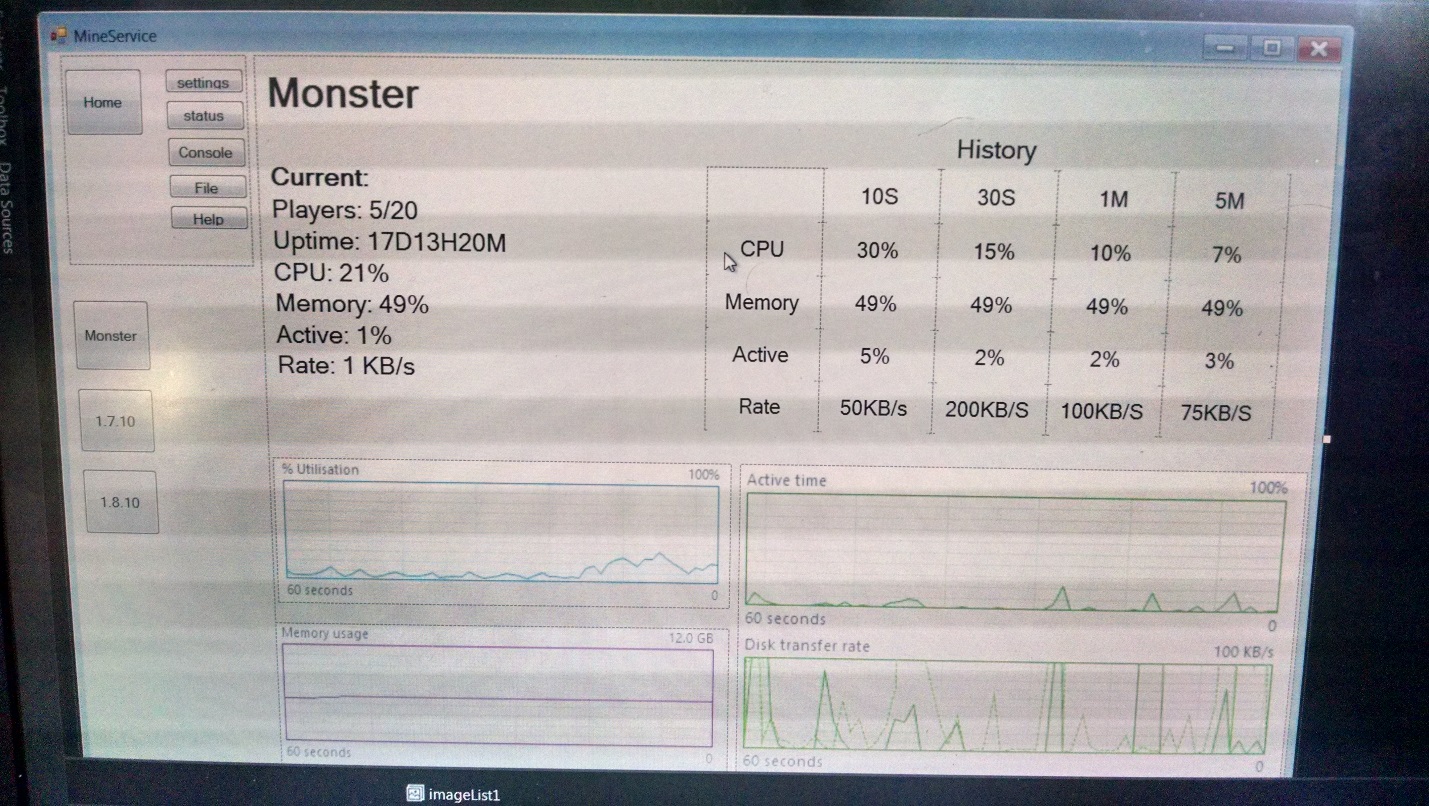
**Plan for Evaluation**

A couple clients that currently pay CJ were given the prototype designs. The process of them using the prototypes was recorded and put into a video file. We hoped to accomplish the layout of the user interface and obtain feedback on the display of the analytical data of the servers. We were unsure especially of which side the tabs should be on for displaying the other server’s data. In addition, we hoped to obtain feedback from the where the commands should be issued.

**Result for Evaluation**

Based on the feedback, we saw that the “file/help” bar should be removed and create a new tab called “home” that this will integrate in with. The elements under the current overview tab will also be under the “home” tab. Overall the feedback was good, just requested a few things being moved around the UI. They also said the first prototype would be good for the desktop application.

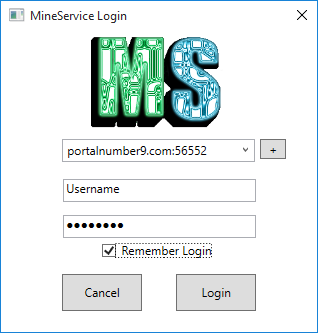
**Changes to Prototype**

****

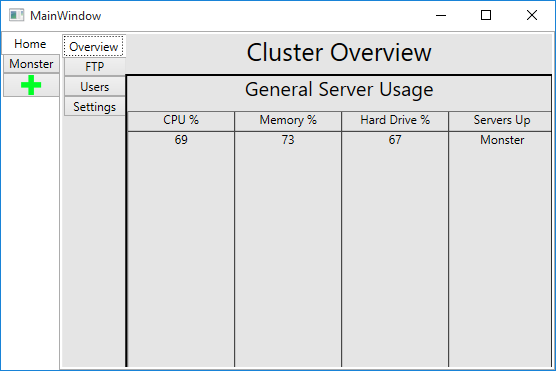
Home tab now involves file and help and the place for entering commands is in the home tab as well.

**High-Fidelity Prototype**

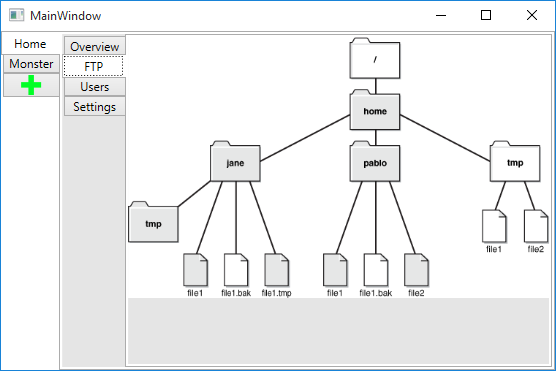
This is the first window you will see. General Login info that can store the servers you connect to.



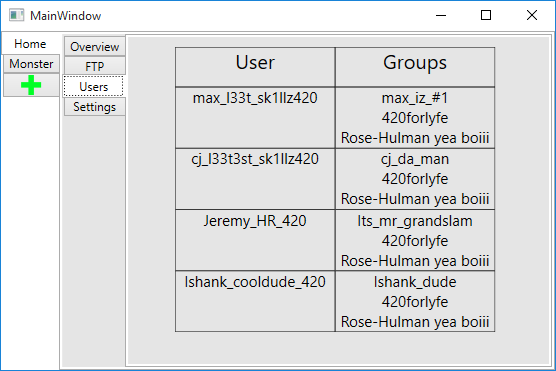
This is the window you will see after logging in. General Information about the cluster



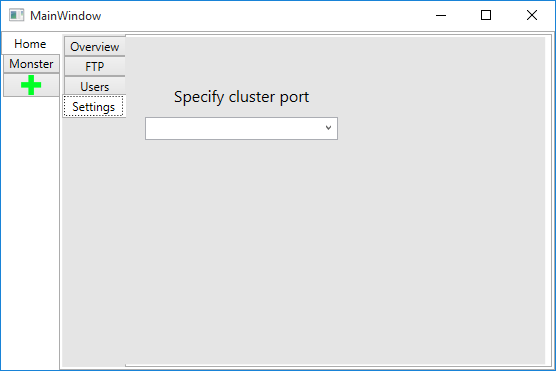
This is the FTP: Design still in progress might go to more of a text based solution. Waiting from response from test clients.



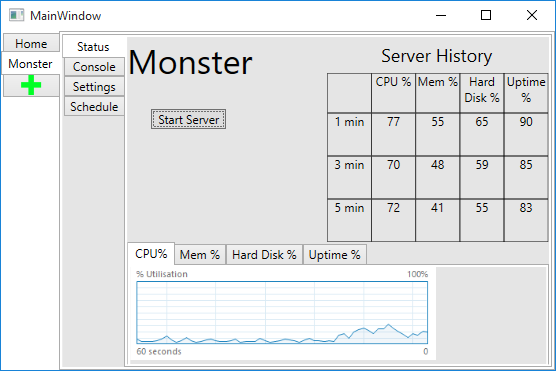
This is basic user control to see who has access to what and in what group.



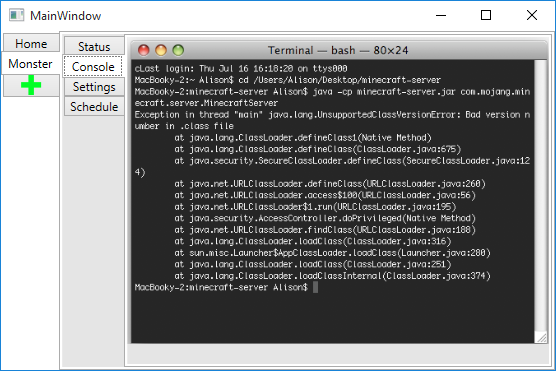
This page will house cluster settings as they are needed. Currently only port is needed, but latter if we extend to additional features this will be needed.



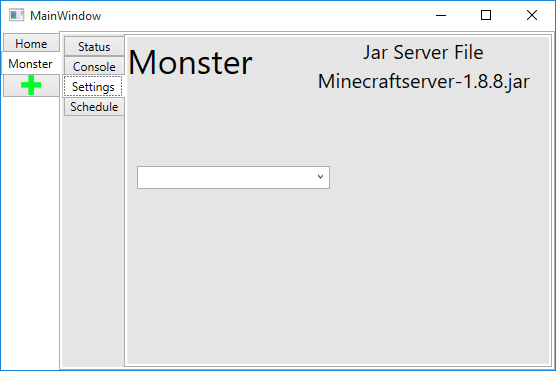
This is the default page for a specific server.



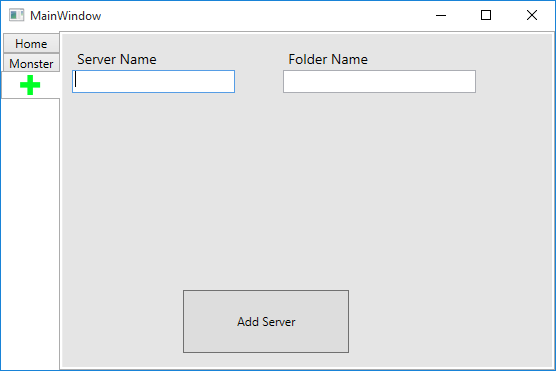
This is the console window for a specific Minecraft server, window still in progress.



Server specific settings. Still in progress.



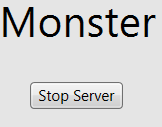
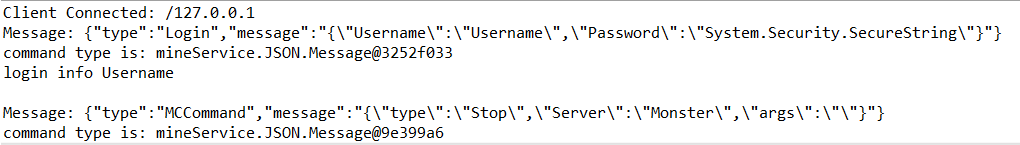
Page to add a new server to the cluster.



**Features Completed**

(note: names listed in approximate order of contribution to feature)

Feature: Add a start/stop server button to the GUI, send the correct message to the server, and make sure the server is receiving the message in the correct format.

Conditions of Satisfaction

* Acknowledge that I don’t have to use terminal to manage my servers

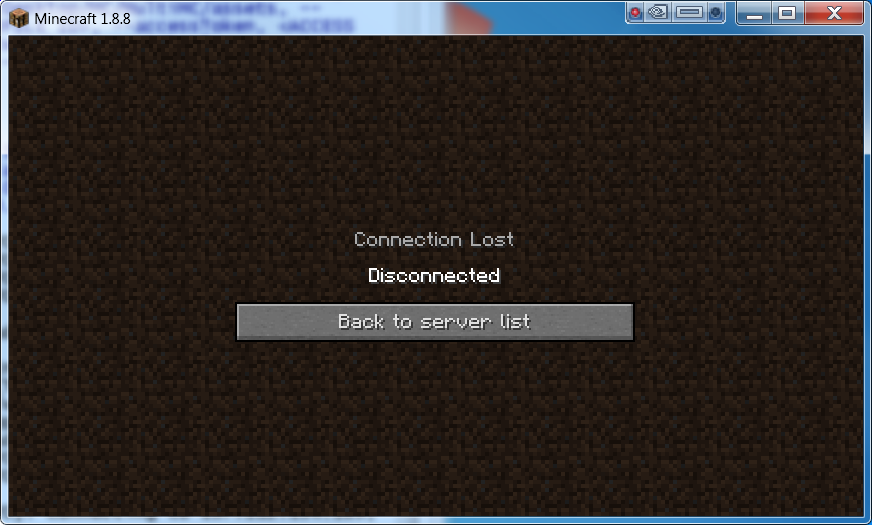
Contributors: Ishank, C.J., Max, Jeremy

Feature: Handle start and stop requests on the server so that a Minecraft server is started or stopped when requested.

Minecraft client a few second after the start message has been received.



After clicking stop on MineService this is seen on the Minecraft client.



On the MineService Server Console the following is seen:



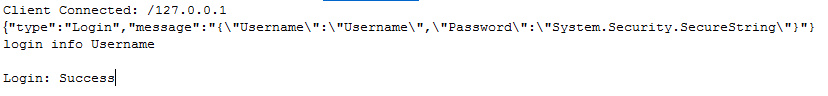
Conditions of Satisfaction

* When a stop message is received the running server is stopped
* When a start message is received the server is started and only if it is not already running.

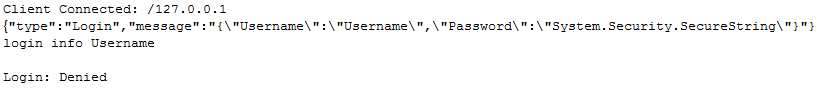
Contributors: CJ, Jeremy

Feature: Users are authenticated against a database to be able to login.

Shows on the server when a user logs in with a good username/password:



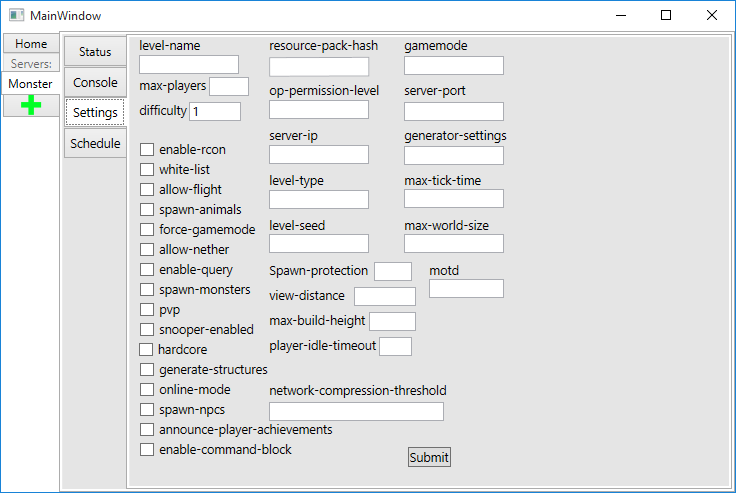
From a bad user/pass combo:



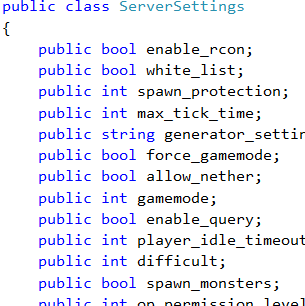
Conditions of Satisfaction

* A login that is stored allows you to connect.
* A login that is not valid disconnects you.

Contributors: Jeremy, CJ

Feature: Improved Settings tab for specific servers with options that represent real possible settings. Also some coding to be used to communicate this information to the server. 

The user can enter settings in this window



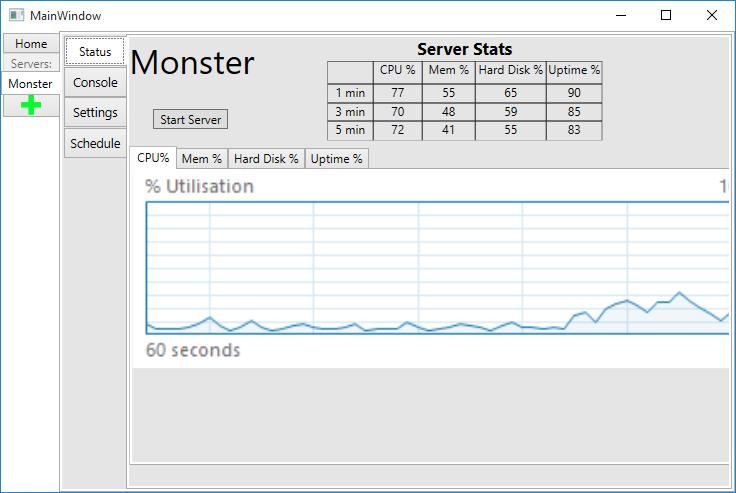
Conditions of Satisfaction

* The user should be able to perform all the actions they want to perform with the given settings.
* The use of the settings should be intuitive enough for a first-time user to use them correctly.
* The settings should be implemented correctly for the in-game experience.

This is part of the C# objects that the user can change in the settings window. These objects will be sent to the server where they will change in-game behavior.

Contributors: Max, Ishank, C.J.

Feature: Used Microsoft Blend to improve aesthetics of the application, and resized elements in windows to make better use of space.

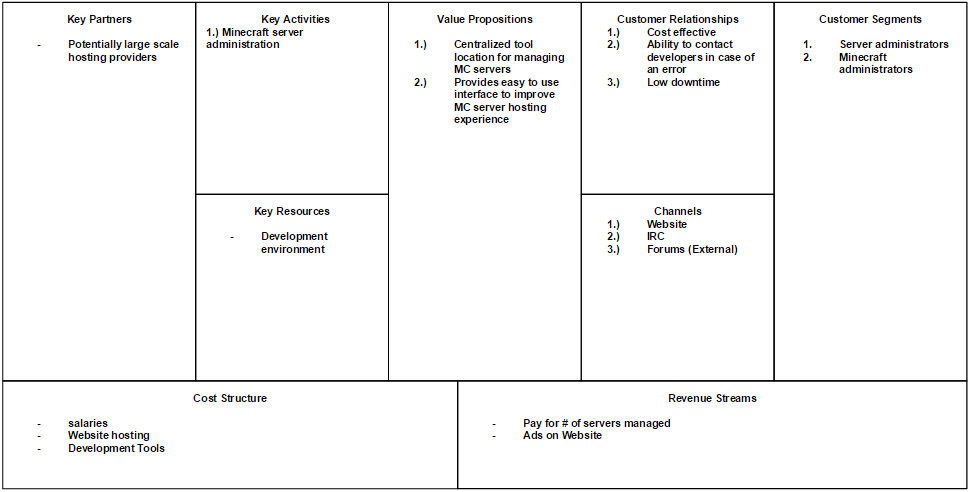


Conditions of Satisfaction:

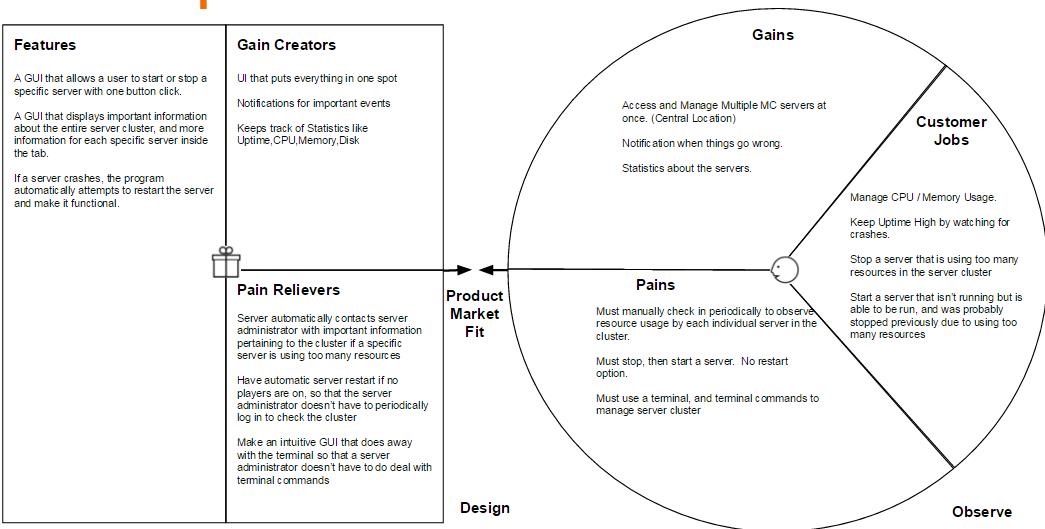
* More appealing and understandable user interface than what existed previously, according to people who didn’t design the UI.

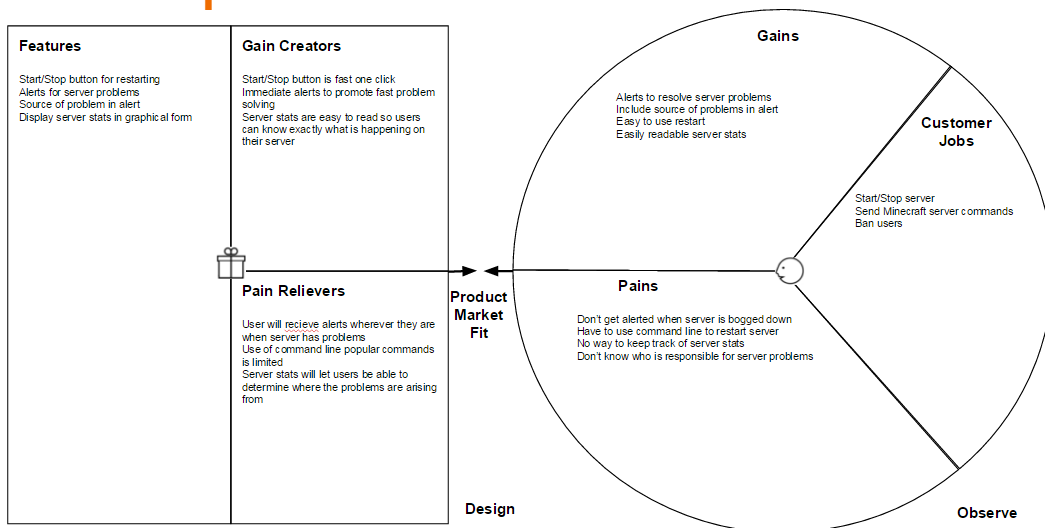
Contributors: Max

**Business Model Canvas**

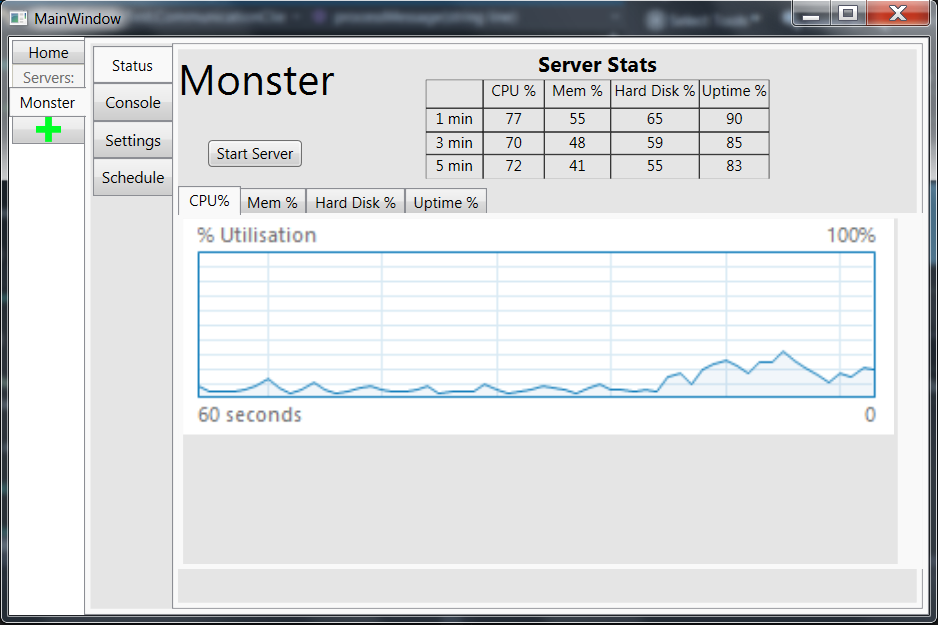
****

**Value Proposition Canvas**

****

****

**Usability Study Improvements**

The feedback from the usability lab was pretty positive and not much had to be changed. The main problem we saw was that clients were having trouble finding the individual server pages, which we plan to improve upon by making a “Servers” heading and placing the servers under that heading. This change is reflected on the left hand side of the application window in the following screenshot.

The usability report can be accessed on the usability lab computer in the CSSE 371 class, 1516, section 2, and in the MineService folder.

**CSSE371 Appendix**

* Items available upon request:
  + Interview documents
  + Usage Diagrams
  + Initial Concept Pictures and Notes
  + Notes Taken on User usage

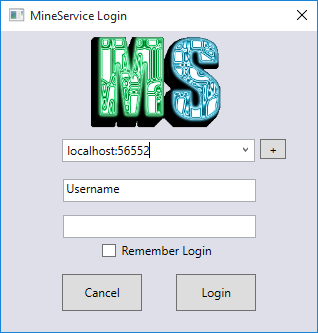
**System Admin Guide**

*Adding a New User*:

To add a new user, simply add the desired username and salted password to the Users.db file.

**User Guide**

*User Login*:



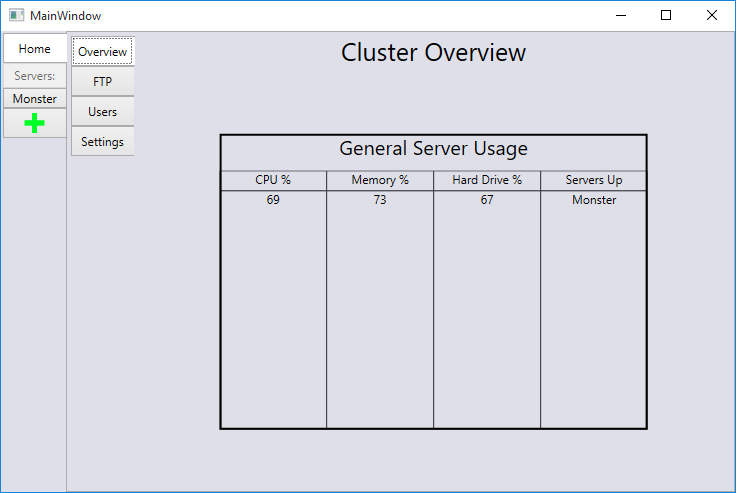
In the first text field put the server in the form of IP:Port (127.0.0.1:55555) that you wish to connect to.

In the next two fields put in the username and password given to you by your administrator.

*Viewing the Home Tab*:

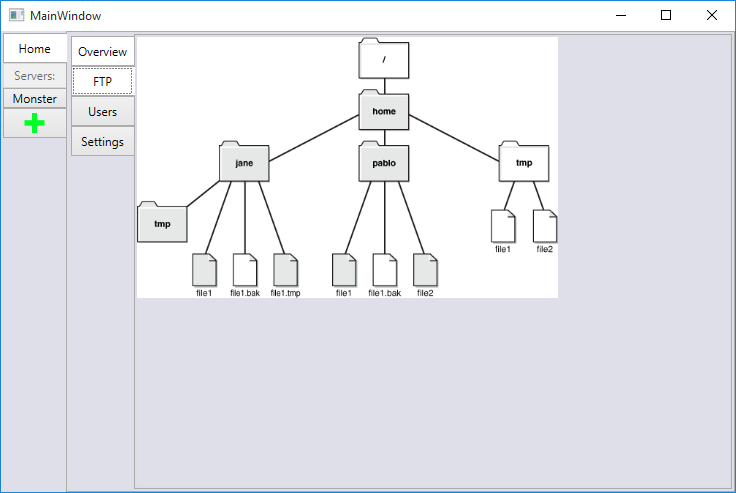
The home tab contains general information about the cluster as long as anything that deals with elements on a cluster level such as settings and users.

Overview (In Progress)



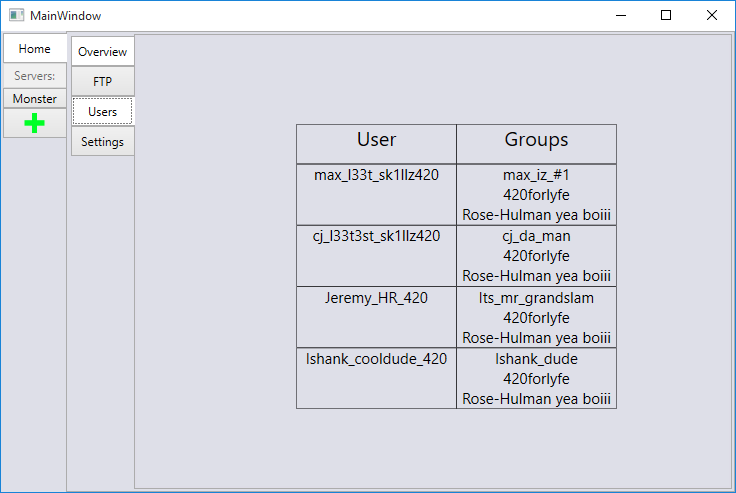
The overview tab is a quick way to see the general status of all the servers on this cluster

*Viewing the FTP Tab (In Progress):*



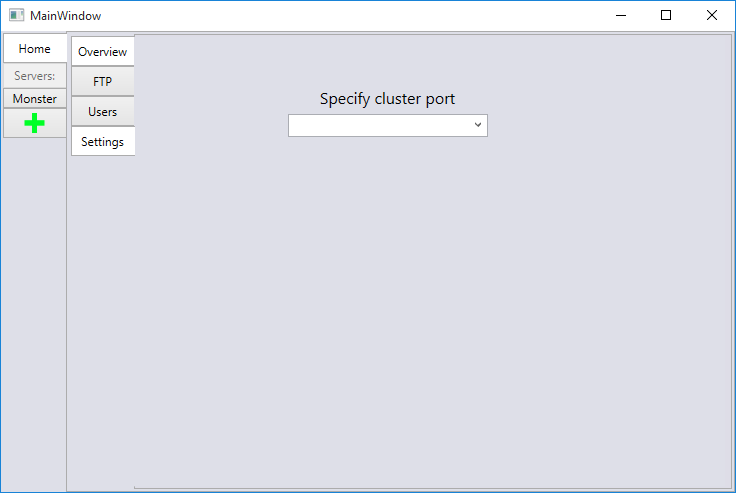
This tab will allow you to make changes to anything in the cluster file tree. This is for experienced users only and is restricted to clients who has full rights on cluster.

*Viewing the Users Tab (In Progress):*



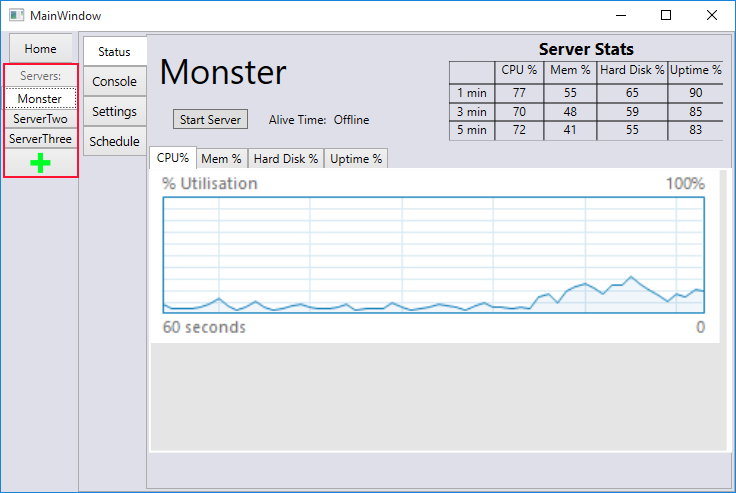
This tab will allow you to add and remove users as well as change their permissions.

*Viewing the Settings Tab:*



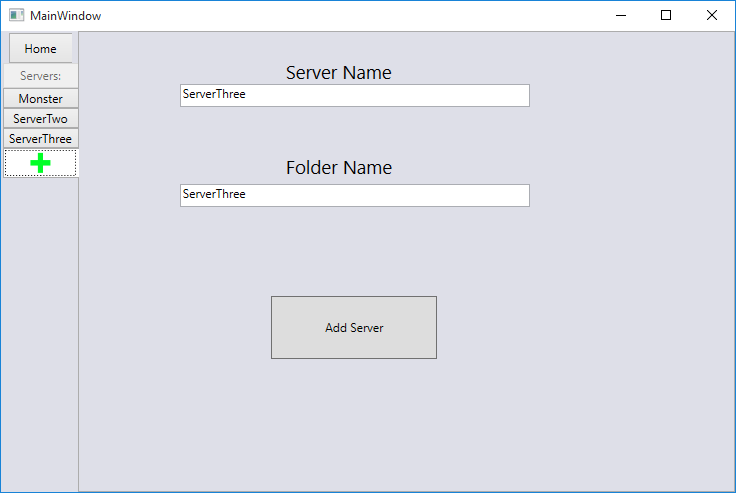
This tab contains the settings for the cluster. Currently only setting is the port number.

*Viewing the Server Tabs:*



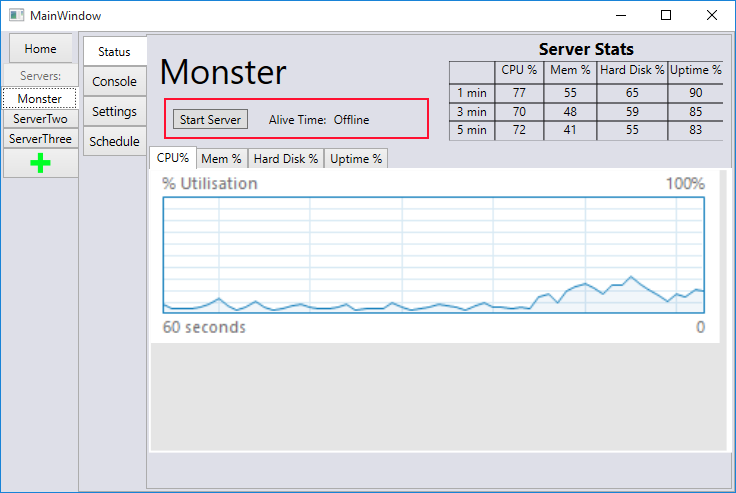
The area marked in red is where any servers on this cluster will show up. The green + is where new servers are added.

*Adding a New Server Tab:*

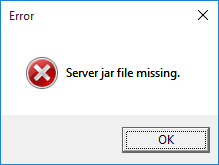


Both the server name and folder name must be unique with what is on the cluster. After putting in two names and hitting add server it will show up with the other servers automatically.

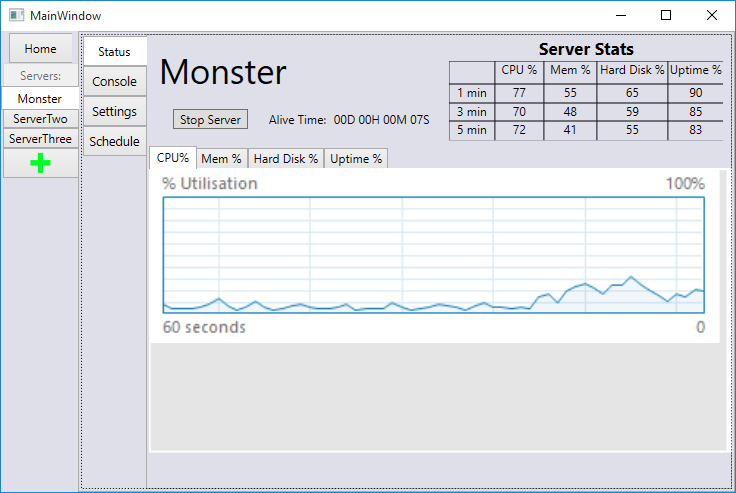
*Viewing a Server:*



Clicking start server will start the underlying Minecraft server.

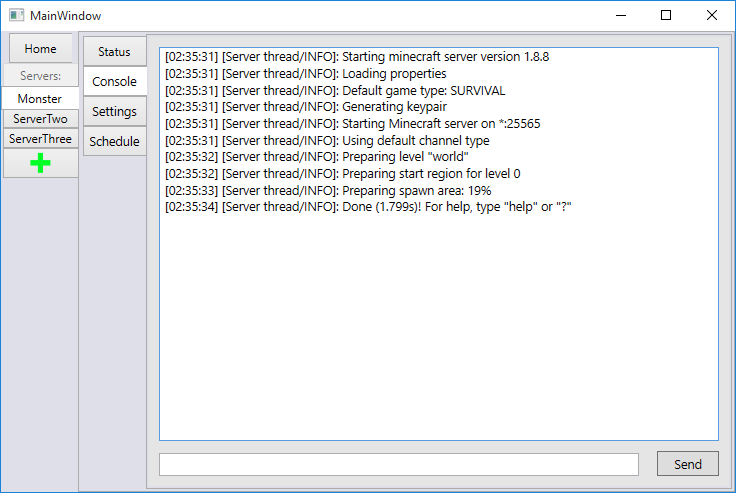


Error will appear if there is no jar that matches the name set in the server settings.



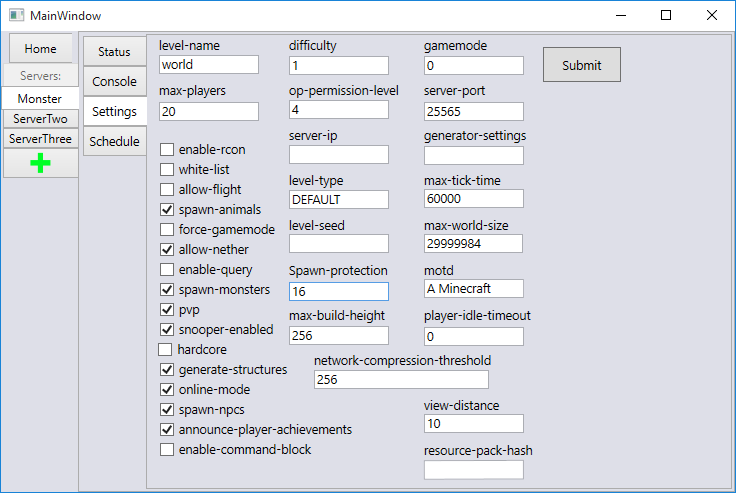
After hitting “Star Server” it will tell the cluster to start the server. If it does start it will change to stop server and start showing the uptime.

*Viewing the Console:*



This screen is a direct pipeline to the normal Minecraft Console. Anything inputted in the bottom will be feed directly to the Minecraft server upon hitting Send.

*Selecting Server Settings (In Progress):*



On this page is all the normal settings found in the server.properties file in the Minecraft root directory.

Schedule (Not yet Developed)

Will allow for automatic restarts based on a time schedule.

Installation / Configuration

Run setup.exe from where it