

# **UI Navigators**

**Universal Media Server**

Aidan Leavy - 18838820

Lachlan Mackenzie - 19154190

Alex Mcleod - 19493178

Brendan Scarfone - 18880708

**HUMAN COMPUTER INTERFACE (ICTE3002)**

**Assignment 1 Report**

**Curtin University**

# Table of Contents

<b>1.0 Introduction</b>	3
<b>2.0 Background</b>	4
2.1 Program Description	4
2.2 Design Purpose	5
2.3 Team Members	5
2.4 Contribution Table	5
<b>3.0 Design Process</b>	7
3.1 User Involvement	7
<b>4.0 Requirement Elicitation</b>	8
4.1 Target Demographic	8
4.2 Survey Results and Analysis	8
4.3 Personas	10
4.3.1 Bill	10
4.3.2 Greg	10
4.3.3 Gary	11
4.4 Competitor Analysis	11
4.4.1 Serviio	11
4.4.2 Plex	12
<b>5.0 Requirement Specification</b>	14
5.1 Functional Requirements	14
5.2 Non-Functional Requirements	14
5.3 User Stories	15
5.3.1 Bill	15
5.3.2 Greg	15
5.3.3 Gary	15
5.4 Use Cases	16
5.5 Hierarchical Task Analysis	17
<b>6.0 Prototypes</b>	19
6.1 Aidan Leavy	19
6.1.1 Low-fidelity Prototype	19
6.1.2 High-fidelity Prototype	24
6.2 Lachlan Mackenzie	26
6.2.1 Low-fidelity Prototype	26
6.2.2 High-fidelity Prototype	28
6.3 Alex McLeod	33
6.3.1 Low-fidelity Prototype	33
6.3.2 High-fidelity Prototype	41
6.4 Brendan Scarfone	50
6.4.1 Low-fidelity Prototype	50
6.4.2 High-fidelity Prototype	53

<b>7.0 Implementations</b>	56
7.1 Aidan Leavy	56
7.1.1 Implementation Discussion	56
7.1.2 User Involvement	63
7.1.3 Test Cases	64
7.2 Lachlan Mackenzie	66
7.2.1 Implementation Discussion	66
7.2.2 User Involvement	69
7.2.3 Testing	72
7.3 Alex McLeod	73
7.3.1 Implementation Discussion	73
7.3.2 User Involvement	74
7.3.3 Test Cases	76
7.4 Brendan Scarfone	80
7.4.1 Implementation Discussion	80
7.4.2 User Involvement	85
7.4.3 Test Cases	86
<b>8.0 Evaluation</b>	87
8.1 Meetings	87
8.2 Milestones	87
8.3 Risk Management	87
8.4 Problems Encountered	88
8.5 Conclusion	88
<b>9.0 References</b>	8989
<b>10.0 Appendix</b>	90
10.1 Initial User Survey Responses	90
10.2 Meeting Minutes	92
10.2.1 05/03/2020	92
10.2.2 26/03/2020	93
10.2.3 16/04/2020	94
10.2.4 14/05/2020	955

# 1.0 Introduction

Contributor(s): Lachlan

This report documents the entire process for designing and developing a Graphical User Interface (GUI) for a media server program, Universal Media Server (UMS). As such, this report includes the method of eliciting our requirements for the project, four low and high fidelity prototypes were then developed mostly independently based on the requirements. These four prototypes were implemented via JavaFX into a fully functional GUI for our chosen program. These implementations were thoroughly tested to be confident that they were abiding by the project requirements. Nearing the end of this document, the entire design process is evaluated in regards to meetings, risk management, milestones and problems.

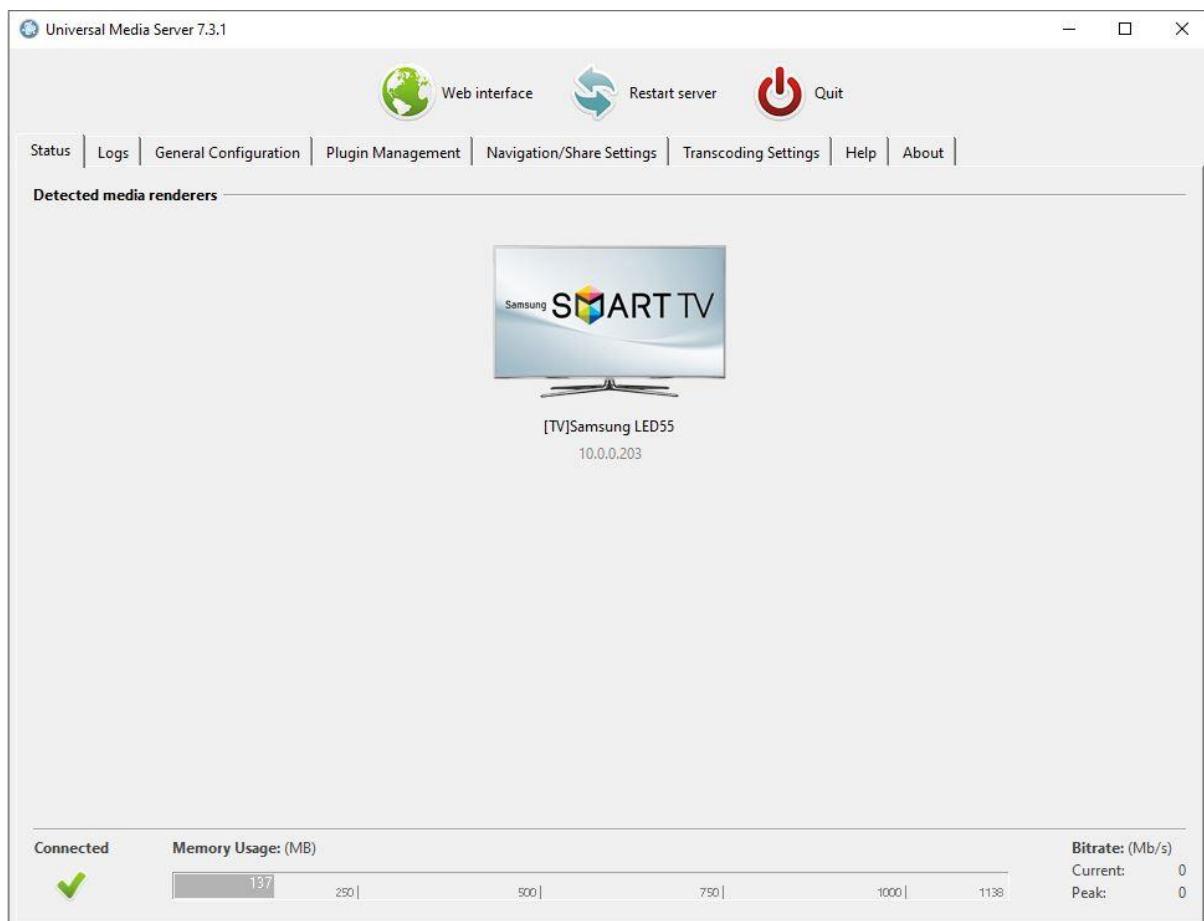
# 2.0 Background

## 2.1 Program Description

Contributor(s): Brendan

The program we have chosen to design a GUI for is Universal Media Server (UMS). UMS is a DLNA compliant UPnP Media server making digital media including video and audio available on multiple devices. The program is written in java, is open source and was based on the “PS3 Media Server” but has been upgraded to work with almost all available media devices. The program suffers from its poor user interface and overly complex settings with the only explanation or information on these settings being buried inside the very long help section. The program does have tooltips for some buttons, but it would make more sense for them to implement more tooltips for these settings.

The program itself is a window with a series of tabs on it that are used to display the relevant information. There is a large ‘Quit’ button above the tabbed area which at first glance seems redundant as the window itself has the ‘X’ to close the window. The ‘X’ only quits the program but not the HTTP server. The interface is not clear on the difference between these two methods of closing the program. The more modern style of the three buttons at the top of the main screen are quite out of place when compared to the rest of the application which is quite reminiscent of Windows 95, making the program feel like it has tried to be modernised, albeit rather poorly.



The majority of the program uses a light gray background, except for the help tab which uses a white background this is because the program uses html documents for the help tab. This is jarring and inconsistent with the rest of the program. The help menu has links to several different pages that are connected through 'next' and 'previous' buttons with an option to return to the help menu, though all of these controls are at the bottom of the page, meaning you have to scroll all the way down to change pages. This can be annoying to deal with as some of the pages are quite lengthy.

## 2.2 Design Purpose

Contributor(s): Brendan

The main purpose in redesigning this application is to make it more appealing and satisfying to the user. It will also be formatted in a user-friendly manner so that a user can perform tasks and find options with ease.

For each of the different tabs available we plan on making icons relating to them as well as making the tabs themselves larger. This along with a lot more colour in the interface would make the program as a whole more attractive and vibrant to the user, resulting in more users using the program.

Some windows available have little to no explanation as to what they do, hence including information when you click on them or when you hover over them would be helpful to users new to the program. We could also include a welcome screen when a user first opens the application. This window will explain to the user how to use the program as well as describe the importance of the different tabs/windows available.

We also want to provide better user feedback when selecting different options. For example, when the user hovers their mouse over a button, we could allow for an animation to occur such as a change in colour. This feedback provided to the user will in turn make the user experience more satisfying. We could also include an outline for buttons that currently do not have one to provide more clarity to the user.

## 2.3 Team Members

Contributor(s): Brendan

- Aidan Leavy - 18838820
- Lachlan Mackenzie - 19154190
- Alex Mcleod - 19493178
- Brendan Scarfone - 18880708

## 2.4 Contribution Table

Member	Sections
Aidan Leavy	5.1, 5.2, 5.3, 5.4, 5.5
Lachlan Mackenzie	1.0, 5.1, 8.1, 8.2, 8.3, 8.4, 8.5, 10.2
Alex McLeod	4.1, 4.2, 4.3, 4.4, 10.1

Brendan Scarfone

2.1, 2.2, 2.3, 3.0, 3.1, 8.4

# **3.0 Design Process**

Contributor(s): Brendan

The software lifecycle chosen for this project is the agile model. This model works by breaking down the product into cycles which focus on small incremental changes. This model emphasizes interaction, as the customers, developers and testers work together throughout the project. With this project the small incremental changes allow users to respond to simple things such as changing the program's theme or layout before they are permanently changed which gives the users the choice of the direction that the product is taking. With something like a GUI change, the user's response is very important. If the user does like the current design, then changing it could put some people off using it so small changes with user feedback is the best way to slowly introduce them to the changes.

## **3.1 User Involvement**

Contributor(s): Brendan

The best way that I see to include users in the design process would be adding a beta type application with the newer updated GUI and get the users impression of the changes. The application as it does use a beta for users that have an interest in using any updated features, this would be a good way of either slowly improving the GUI with e.g. some simple colour theme changes or just adding a dark mode to see how the more invested users will react. These updates could also come with a survey to get impressions or as they currently do just post it to the UMS forum where users can leave feedback on updates in the comments.

# 4.0 Requirement Elicitation

## 4.1 Target Demographic

Contributor(s): Alex McLeod

The main demographic of our application is people who want to stream their media to different devices in their home and store their media in one place. This includes people with an understanding of media servers who want an easy way to customize their own media server and people with little to no knowledge of media servers who just want a way of being able to quickly stream their media to their devices. This demographic is proved through the results of the user survey shown and analysed in the next section of the report. The survey used can be found in the appendix of the report.

## 4.2 Survey Results and Analysis

Contributor(s): Alex McLeod

Survey Results:

Survey conducted with 100 users of the Universal Media Player Application

*Question 1: What age are you?*

Results: **20%** aged 10 – 19, **60%** aged 20 – 39, **%10** aged 40 – 49 and **%10** aged 50 or older.

*Question 2: What is your occupation?*

Results: **20%** primary school/high school students, **30%** university students, **10%** computing students, **20%** full time worker, **20%** involved in full-time work related to computing.

*Question 3: On average how many hours would you spend watching videos in a day?*

Results: **20%** less than 1 hour, **50%** between 1 and 3 hours, **30%** more than 3 hours.

*Question 4: Would you say that you have a good understanding as to how media servers work?*

Results: **30%** yes and **70%** no.

*Question 5: Did you seek help from another individual to get your media server working?*

Results: **60%** yes and **40%** no.

*Question 6: Are you able to easily find the settings you want to change?*

Results: **30%** yes and **70%** no.

*Question 7: Do you believe that the look of the application is appealing?*

Results: **10%** yes and **90%** no.

*Question 8: Is the application satisfying to use?*

Results: **20%** yes and **80%** no.

#### Survey Analysis:

From this survey, we are able to get a better understanding of the demographic associated with this application. It also gives us a better idea as to what is wrong with the current interface, allowing us to change or improve different aspects for our new design.

From the first question we can see that there is a large majority of young adults that use the application with 60% of users being between 20 and 39. From this we can assume that the majority of users of the application have some familiarity with current interfaces and technologies. So, the basics of how to use a general interface should be familiar to them. This also allows us to assume that if we use common icons that are universal to many modern interfaces in our new design, they will most likely be familiar with them and their meanings. So, using common icons would be beneficial in helping our users identify different screens.

From question two we can see that the occupation of the users of our program are quite varied with the highest percentage being university students at 30%. This hence shows how our design must accommodate a variety of users. For those people with a lot of computing and media server knowledge they will most likely want the ability to change a number of server settings so that the server can run the way they want. Hence including different settings options is important to allow this customization. From question 6 we found that 70% of users found it hard to find the settings they wanted to change. Hence it will also be important to set up the menus in an orderly and intuitive manner so that users who want to customize their server can do so more easily. Changing the settings on the old interface would have been challenging at times as there was little order to the settings options.

From the varied occupation of users, we can also assume that many people have little to no knowledge as to how media servers work to stream their media. This being supported by the answers to questions 4 and 5. With 70% of users saying that they don't really know how media servers work and 60% saying that they needed to find help from other users in order to get their media server working. It shows how the original interface gave little guidance to new users of the application. Hence it will be important to make our new interface user friendly so that all users can get their server working. Including things such as explanations for each screen, tooltips and a more intuitive help screen will be vital in ensuring all users can use the application.

From the survey we also found that increasing the visual appeal and satisfaction when using the application will be important to please users. This is supported through the answers to questions 3 and 7. From question 3 we found 50% of people spend an average of 1- 3 hours watching videos in a day and 30% who watch more than three. Users will therefore be opening and closing the application a lot as they stream their media. Hence making the application both visually appealing and satisfying for the user will be important. This is also supported by questions 7 and 8 where 90% of users believe the application is not visually appealing and 80% believing it is not very satisfying to use. Hence to improve the interface visually we will need to add more colour and layout the different elements, so they are not

crammed together. To improve the satisfaction of using the interface we will need to add more user feedback. This may include feedback when pressing buttons such as colour changes or animations.

## 4.3 Personas

### 4.3.1 Bill

Contributor(s): Alex McLeod

Bill is a full time computing student. He spends most of his time during the week programming for his university assignments. In his free time, he likes to create his own programs without restrictions on what he can and can't include in them. As well as this he enjoys watching his favourite films.

Background:

- Male, 21 years old
- Occupation: third year computing student

Motivations:

- He wants a highly customizable media server application that allows him to run the server the way he wants.

Frustrations:

- An application that is too simple and does not allow him to customize all the settings that he wants
- Not being able to find settings he wants to change

### 4.3.2 Greg

Contributor(s): Alex McLeod

Greg is a full-time manager at a popular business firm. He works 8 hours a day during the week and finds himself extremely stressed out. In his spare time, he likes to destress by watching his favourite television shows. He wants a way of being able to easily watch his favourite shows on any of his media devices without any hassles.

Background:

- Male, 35 years old
- Full time manager at a business firm
- Little computing knowledge

Motivations:

- Wants a way of easily watching his favourite television shows from any of his devices.
- Wants an application that is stress free and allows him to easily set up his own media server.
- Wants an application that provides as much help as possible for new users.

Frustrations:

- Spending hours trying to figure out how an application works.
- Not being able to find the help he is looking for.

### 4.3.3 Gary

Contributor(s): Alex McLeod

Gary is a high-school student. The money he earns from his casual jobs he loves to spend on flashy new computer games with the best graphics. He doesn't like video games that look old or have poor graphics. When he is not playing his new video game's he is watching his favourite action movies.

Background:

- Male, 17 years old
- Highschool student

Motivations:

- He wants a clean and modern looking media server application that will allow him to stream his favourite movies to his different devices.
- Wants to be able to watch his movies as soon as possible.

Frustrations:

- He hates old looking applications.
- He wants to be able to quickly stream his videos
- Hates wasting time.

## 4.4 Competitor Analysis

### 4.4.1 Serviio

Contributor(s): Alex McLeod

Serviio is a media streaming server application developed by Petr Nejedly. It is currently one of the most popular media streaming servers.

Advantages:

- Includes several icons relating to each menu tab on the left. This adds to the visual appeal of the interface and informs the user of each tabs use.
- It uses a range of different colours that complement each other further adding to visual appeal.
- When tabs are hovered over, they appear darker and when a tab is clicked its colour changes to orange. This hence provides feedback to the user which makes using the menu more satisfying for the user.

- Some tabs are also expandable and split into sub sections. This allows the user to easily find certain options that are related to each other. It also acts to split up information so that one tab does not become too cramped with information.
- When a tab is selected the screen related to that tab provides a small paragraph explaining the purpose of that screen. In turn providing help to users new to the program

Disadvantages:

- The help information provided on each screen can be hard to see particularly for those with poor vision. This is because the light grey text against the white background causes the text to be hard to read.
- There is no about tab to provide information about the creators, supported devices or important links etc.
- There is no dedicated help tab to provide more in-depth information on how the application works.

#### **4.4.2 Plex**

Contributor(s): Alex McLeod

Plex is a user-friendly media streaming software created by Plex Inc that allows you to quickly and easily stream your media to different devices.

Advantages:

- Extremely user friendly as it walks you through multiple screens to get started with the application. It hence allows you to quickly get your media server set-up with ease.
- It uses a great colour scheme that is visually appealing. It mainly uses dark grey and black colours with yellow highlights. Doing this allows important parts of the interface such as the current tab you are on or notifications to be easily seen by the user as the yellow highlights pop out to the user.
- Smart use of icons in place of text that give the application visual appeal. By having familiar icons that are used in most applications the user can easily identify what tab they are currently on. By replacing text with icons, it saves space on the interface giving it a cleaner look.
- It gives you the ability to pin your most used tabs to the main screen. This adds to the efficiency and customizability of the interface as depending on the user they can choose which tabs are most important to them and pin them for easy access.
- Smart design of the settings menu with a side scroll pane that contains sub-headings for each type of setting allowing the user to easily find the setting they wish to change.

Disadvantages:

- Has unnecessary tooltips on the left tabs. They are redundant as they do not give any extra information but just repeat the name of the tab. The white tooltip box also does not fit with the rest of the application's design.
- Upon starting the application, it suggests a lot of popular movies that have been played on plex. These movies are random and unrelated to the user's preferences.

Hence many users may not want their home screen instantly filled with movies that aren't theirs, creating unnecessary clutter.

- The help button is hidden in an expandable menu at the top right of the screen. It is important for help information to be easily accessible for the user. It should be placed on the home screen or in a menu obvious to new users.

# 5.0 Requirement Specification

## 5.1 Functional Requirements

Contributor(s): Aidan, Lachlan

1. Program should display a home/status page
2. Program should display the detected media renderers
3. Program should allow the user to play media by clicking on a renderer
4. Program should allow the user to open the web interface to the server
5. Program should allow the user to restart the server
6. Program should allow the user to shut down the server
7. Program should display the currently used RAM
8. Program should display the current bitrate of all transfers
9. Program should display the log file
10. Program should allow the user to search the log file via search terms
11. Program should allow the user to filter the log file via the log level (DEBUG etc.)
12. Program should allow the user to open the log file in their default viewer
13. Program should allow the user to pack/zip the log file
14. Program should display settings page(s)
15. Program should allow the user to change its settings
16. Program should have the settings persist
17. Program should display an about page
18. Program should have the documentation/help available to the user
19. Program should display links related to it
20. Program should allow the user to open hyperlinks
21. Program should display the current version
22. Program should display the UMS logo
23. Program should display tooltips to explain certain functions/settings
24. Program should have visual feedback depending on what the mouse is over
25. Program should allow the user to add a folder for sharing
26. Program should allow the user to delete a folder for sharing with confirmation
27. Program should allow the user to monitor a shared folder for its played status
28. Program should respond to resizing to some extent

## 5.2 Non-Functional Requirements

Contributor(s): Aidan

### Performance:

- It should take no more than 0.5 seconds for the program to switch between different windows after clicking on a button in the menu.
- It should take less than 1 second to open the web interface after the web interface button is clicked.

### Usability:

- The user must be able to access any part of the program's functionality within 4 clicks.
- On average, users should be able to specify all optional log search parameters within 5 seconds after typing the log name in the search bar.

**Reliability:**

- The mean time between program failures that require program restart should be no more than 2 months.
- The mean time between streaming failures that prematurely end a stream from the server should be no more than 1 month.

**Security:**

- The server must not be able to stream to any device without server side user authorization.
- All data that is streamed from the server must be encrypted using Transport Layer Security (TLS) over HTTPS.

**Localisation:**

- Must be available in English US, English UK, Brazilian Portuguese, Chinese Simplified, Chinese Traditional, Czech, Danish, Finnish, French, German, Italian, Japanese, Polish, Romanian, Slovak, Spanish and Turkish.
- Must be able to support additional languages in future updates.

## 5.3 User Stories

Contributor(s): Aidan

### 5.3.1 Bill

Bill is someone who enjoys programming in his spare time and wants a program that he can customize in great detail in order to achieve the best experience possible for his environment.

1. I want to be able to play movies using the program.
2. As someone who is willing to learn what all the settings do in order to achieve the best performance possible, I want a detailed list of settings to customize that control all of the program's functionality.
3. As I want to spend a lot of time manipulating the settings, I want the settings menu to be easily accessible

### 5.3.2 Greg

Greg is someone who spends most of his day at work and wants a simple way to watch his favourite shows in his limited time. He doesn't know much about computers and is often frustrated at the lack of help available in other programs.

1. I want to be able to watch shows using the program.
2. I want to be able to easily create my own server with minimal hassle.
3. I want to be able to easily find a help menu.
4. I want to be able to use this program to watch my shows on any of my devices.

### 5.3.3 Gary

Gary values modern aesthetics and high speed in his program. Wasting time is a huge peeve of his and he values any way to minimise time wasting.

1. I want to be able to watch movies using this program.
2. I want to be able to start playing a movie as fast as possible.
3. I want the program to have a modern design.

## 5.4 Use Cases

Contributor(s): Aidan

### Task: Search for a specific log

1. The user selects the Logs tab.
2. The program changes to the Logs menu.
3. User clicks on the search bar and types a string from the log they are looking for.
4. The user clicks on any of the optional checkboxes that apply to their search request.
5. User clicks on the search button.
6. Program goes through all available logs and filters out all of them that do not meet all the requirements.

### Task: Add local folder to server

1. User clicks on the Shared Content tab.
2. Program changes to the Shared Content menu.
3. User clicks the Add Folder icon.
4. Program opens a menu showing a list of all Folders within the users Document folder
5. User clicks on the folder they want to add and clicks on the Open button.
  - 5.1. If the folder the user wants to add isn't in the menu, they can type in the address of the folders location on the computer into the search bar, or navigate to it by double clicking on a folder to change the menu to that folders contents at which point the program returns to step 4.
6. Program returns to the Shared Contents menu with the folder selected in step 4 appearing in the list of folders that can be accessed by the server.

### Task: Remove local folder from server

1. User clicks on the Shared Content tab.
2. Program changes to the Shared Content menu.
3. User clicks on the folder inside the Folder window that they wish to remove from the server.
4. Program highlights the folder that the user clicked.
5. User clicks on the Remove Selected Folder button.
6. Program removes the selected folder from both the Folders menu and the server.

### Task: Add web content to server

1. User clicks on the Shared Content tab.
2. Program changes to the Shared Content menu.
3. User clicks the Add New Web Content button.
4. Program creates an add new web content window.
5. User clicks the Type dropdown menu and clicks on the option corresponding with the type of content they are adding.
6. User Clicks on the Folders text bar and types the names of the folders they want the web content to be stored under.
7. User clicks on the Source/URL text bar and types the URL of the web content they want to add to the server.
8. User clicks on the OK button.
9. Program closes the New Web Content window and returns to the main program on the Shared Content tab.

10. Program adds the new web content into the web content window.

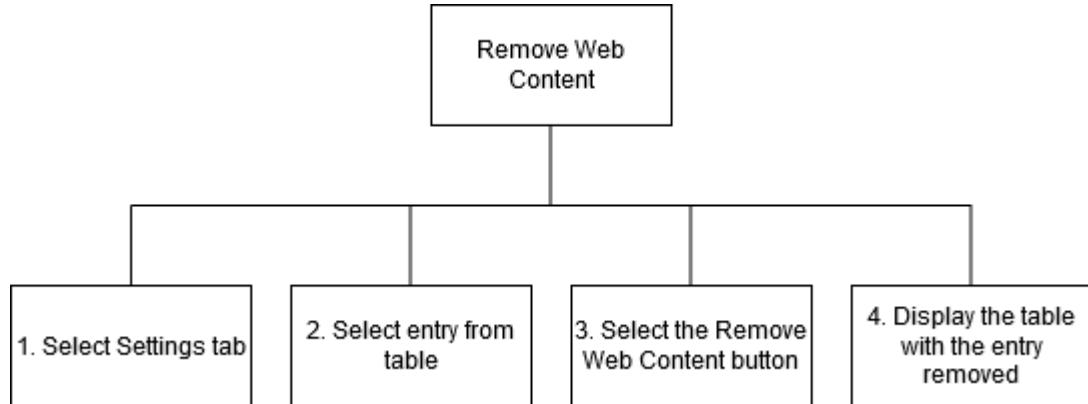
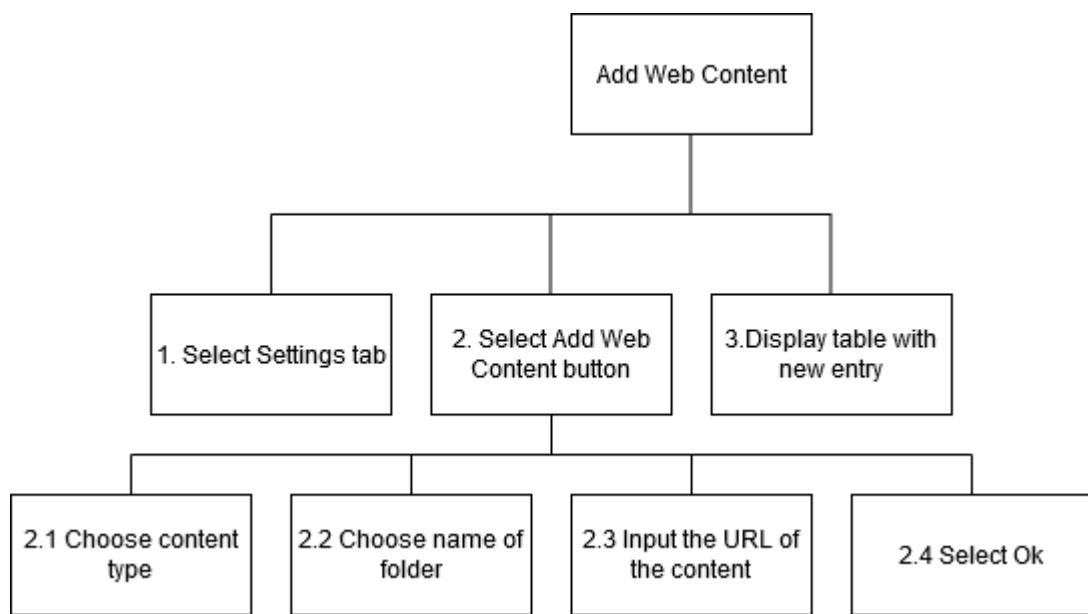
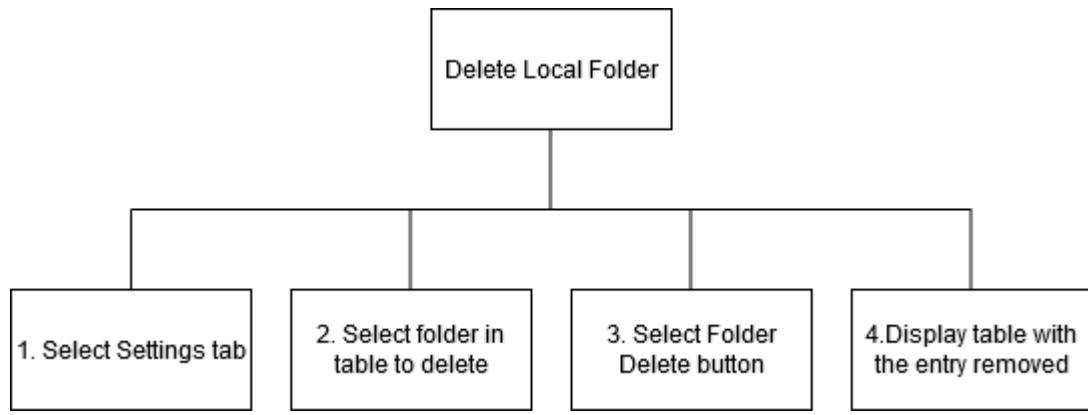
Task: Remove web content from server

1. User clicks on the Shared Content tab.
2. Program changes to the Shared Content menu.
3. User clicks the Remove Web Content button.
4. Program highlights the web content that the user clicked on.
5. User clicks the Remove Selected web Content button.
6. Program removes the selected web content from both the web content menu and the server.

## 5.5 Hierarchical Task Analysis

Contributor(s): Aidan, Lachlan



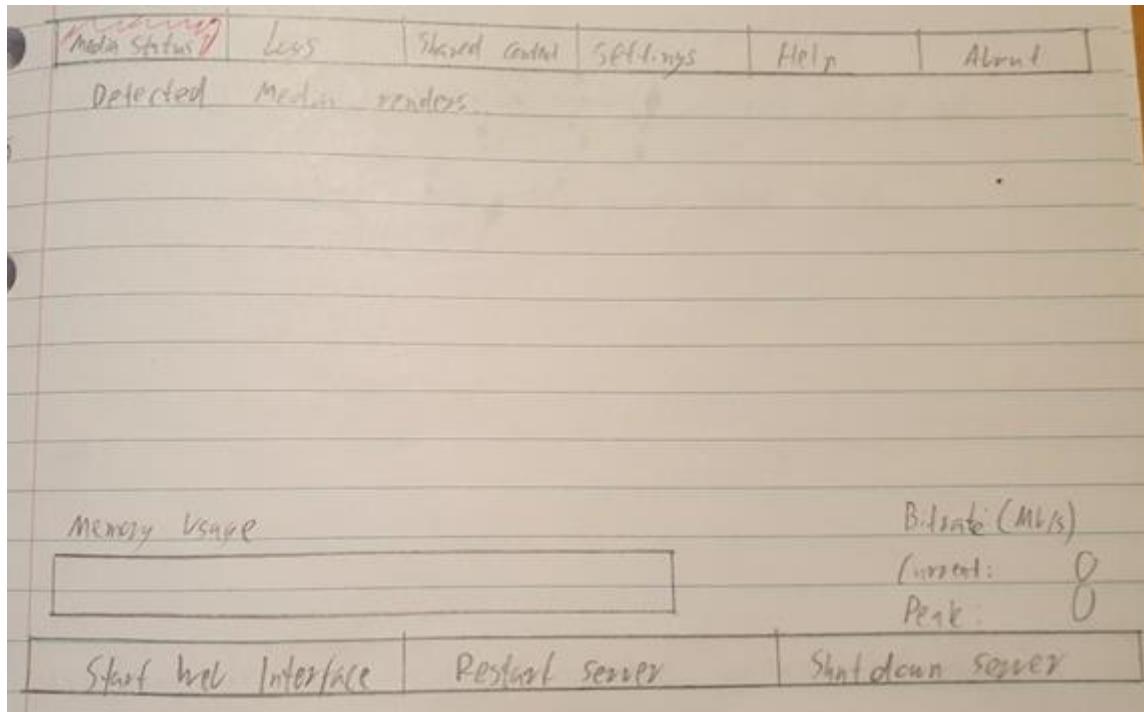


# 6.0 Prototypes

## 6.1 Aidan Leavy

Contributor(s): Aidan

### 6.1.1 Low-fidelity Prototype

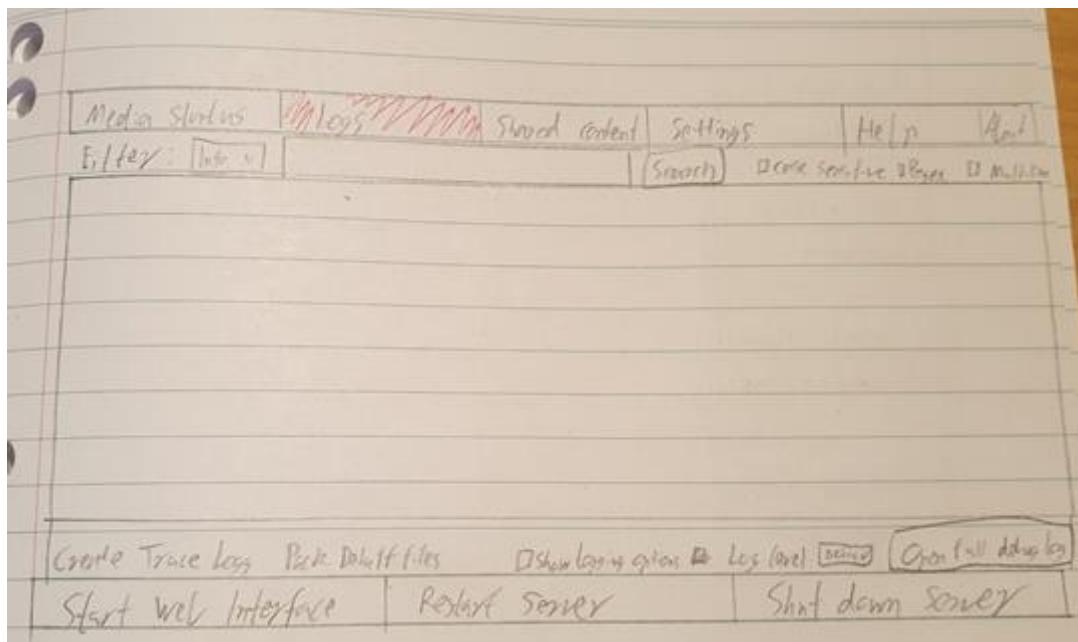


Media Status screen.

This screen displays all of the media renderers (devices the program stream to) that the program has detected, the amount of memory it is using and the current/peak bitrate. In the event that no media renderers are detected, the program will display text that states no media renderers are detected. The memory usage is a progress bar that will fill up based on the percentage of memory currently in use. The bitrate will show the rate of data transfer between the server and its associated media renderers, displaying both the current bitrate and peak bitrate in Megabits per second.

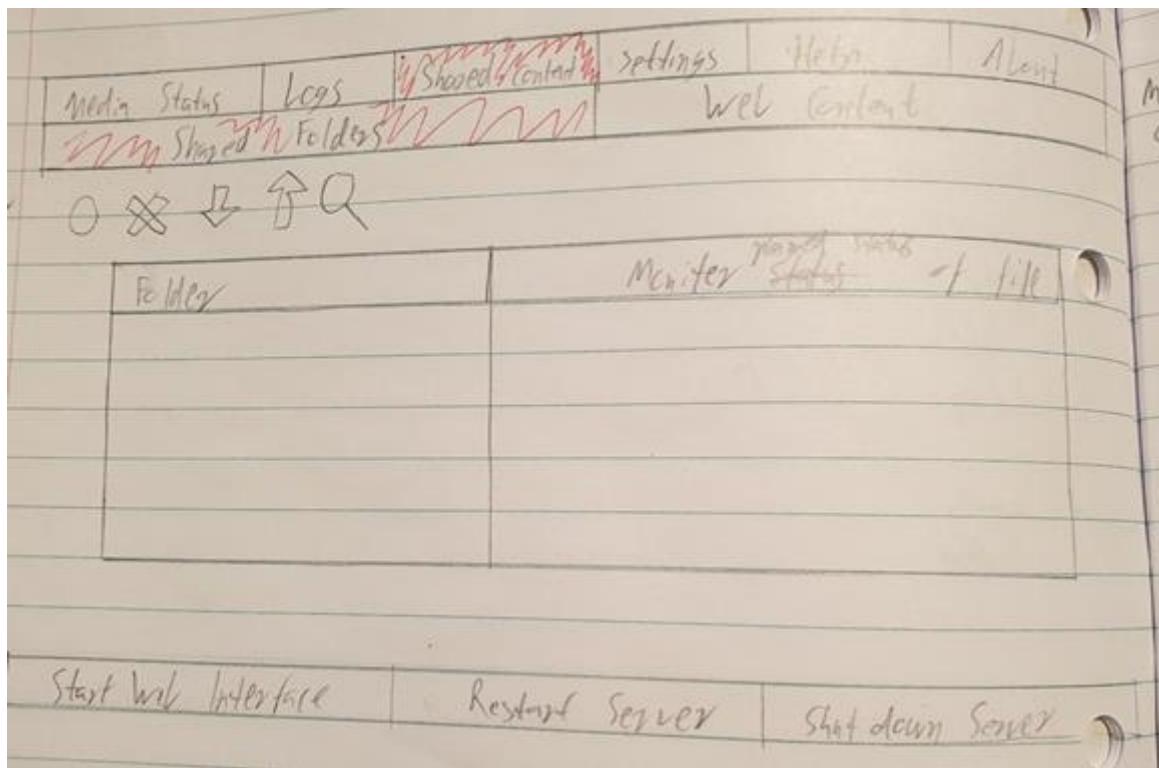
At the top of the screen is a bar with 6 buttons on it. These 6 buttons make up the primary menu bar and are how the user navigates the program. By clicking on a button, the user will be sent to the screen associated with that button, the button will be highlighted red and the user will be able to access that screen's features. Several of these screens have a second menu bar that will be displayed underneath the primary menu bar, called the secondary menu bar, that will allow the user to navigate through that screen's subcategories, however the Media Status screen does not have a secondary menu bar. When the program is first opened, the Media Status screen will be the screen that the program opens to and the Media Status button in the primary menu bar will be red.

At the bottom of the screen will be the final menu bar with options to start a web interface, restart the server and to shut down the server. Clicking Web Interface will open a browser based version of the program in the computer's default browser, Restart Server will restart the server and Shutdown Server will turn off the server. None of these options will affect the program which will continue to run as normal.



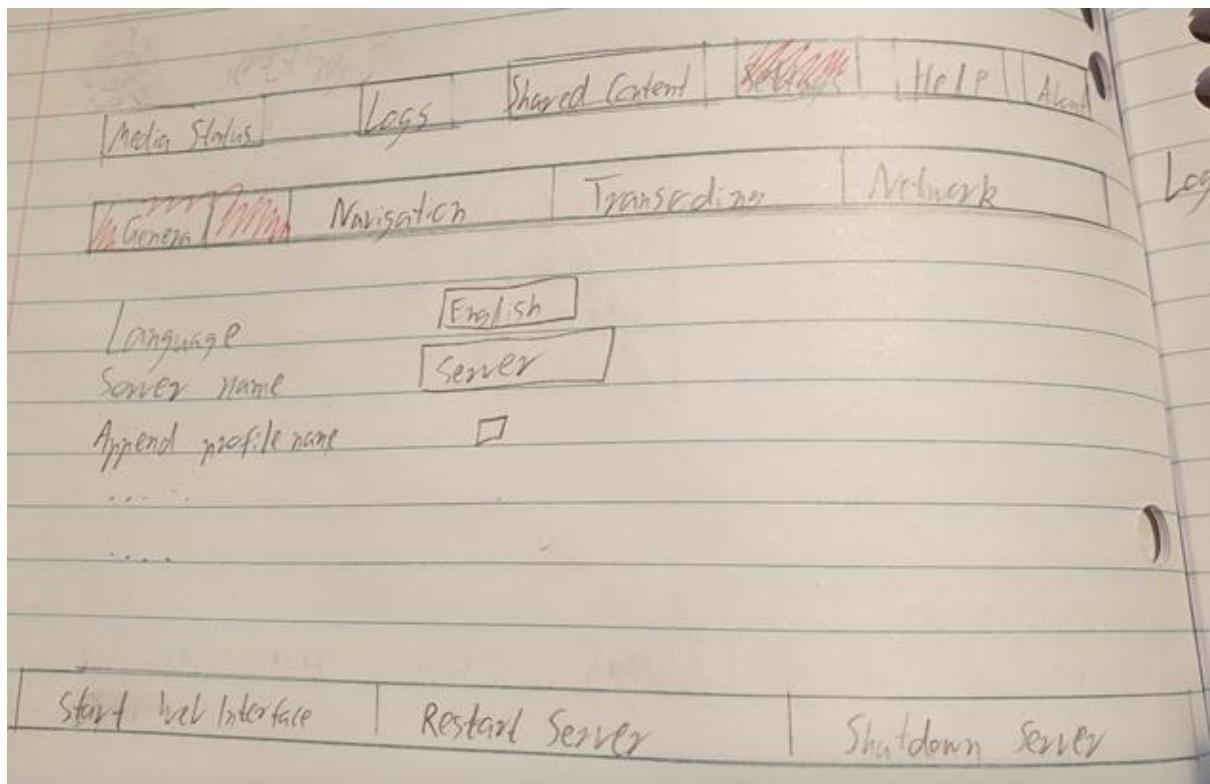
Logs screen

The Logs screen will display a list of logs produced by the server during its operation. This exists so that the user can identify any errors the program is experiencing in order to fix them as well as allowing the user to see what the server did even if nothing went wrong. The logs will be listed in the box in the middle of the screen with a list of search options available above it, underneath the primary menu bar. These options include a dropdown menu to filter specific types of logs (such as error, warning and info) and 3 checkboxes to test for case sensitivity, regex and multiline. Once these options have been configured, the user will type in a search request for the logs they want and the logs that don't meet the specified criteria will be filtered out, leaving only the logs the user requested. At the bottom of the screen, between the log window and the final menu bar, are several options for generating and compressing logs, such as Create TRACE Logs which will cause the program to start producing trace logs, Pack Debug Files which allows the user to compress selected logs into a zip file, Open Full Debug Log which opens a text file containing all the logs and a check box for Show Logging Options which provides even more logging options to the user.



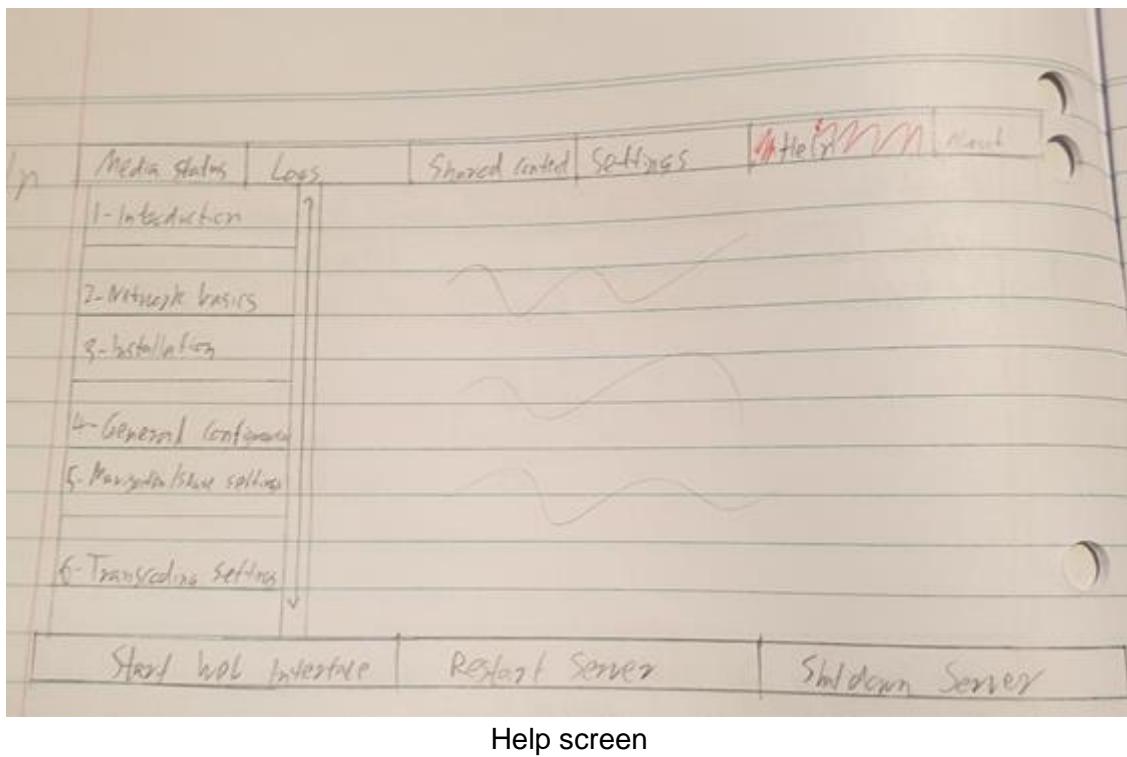
Shared Content screen - Shared Folders

The Shared Content menu is used to manage all of the content hosted on the server. Since the server can handle both content that is stored on the local machine running the server and web content accessed via the server's internet connection, this screen is split into two categories accessed via the secondary menu bar with one button for Shared Folders and one button for Web Content. By default, the Shared Folders button will be selected, and that button will be highlighted red, just like the Shared Content button is in the primary menu bar. In the Shared Folders screen as shown above, there is a window that will display the address of all folders available to the server as well as a checkbox that monitors whether or not the contents of the folder have been played or not. Between the secondary menu bar and the folder window are 5 buttons which are used to manage the folders available in the window. Add folder allows the user to add a new folder to be managed by the server, Remove Selected Folder will remove the folder that is currently being selected by the user in the folder window, Move Selected Folder Down will move the folder that is currently being selected down one place inside the folder window, Move Selected Folder Up will move the folder that is currently being selected up one place inside the folder window and Scan All Shared Folders will scan the folders in the window for content. The Web Content window has similar options, except for the scanning option which is redundant for web content.



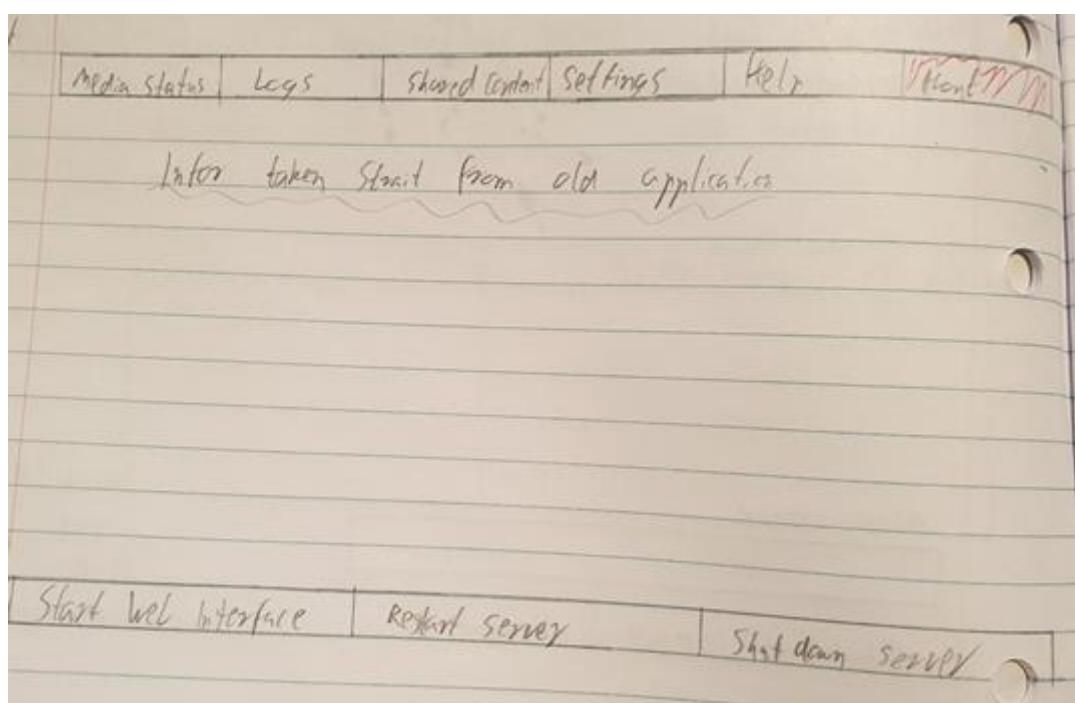
Settings screen - General

The Settings window allows the user to access and modify all the settings the program has. Due to the amount of settings available, they have been separated into four different categories accessible by the secondary menu bar. These categories are General, Navigation, Transcoding and Network. Many of these settings are too complicated for the average user, which is why the General button is selected by default as it contains the most user friendly settings, such as changing the language of the program and changing the server's name. Navigation controls how the user can navigate the server from their media renderer, Transcoding controls how the content is encoded and Network controls how the server maintains its internet connection. The settings are controlled by checkboxes, dropdown menu's text fields and buttons, depending on what the setting is.



Help screen

The Help screen is how the user can navigate the different help pages that come with the program, explaining the most basic of steps such as installing the program correctly and a more detailed explanation of the settings and more complex operations such as installing plugins and using command line arguments. The different categories can be selected from the scroll menu in the right hand corner with each button opening up its specific help menu on the right hand pane. Due to the different help menus having no flow to them and existing as discrete entities, there is no previous or next button to navigate between help menus, meaning the scroll menu is the only way to change the help menu being displayed.



## About screen

The About screen displays the Universal Media Server logo along with several links associated with the development and use of Universal Media Server. The programs current version and build date are also listed.

### 6.1.2 High-fidelity Prototype

Due to time and technical constraints, only two high fidelity prototypes were completed.



Media Status screen

The Media Status screen did not change at all between the low and high fidelity, aside from the background colour which was made black to create a dark theme that is popular among many other programs. There is also a white bar separating both the primary menu bar and the final menu bar from the rest of the window. This was done to make the buttons stand out more from the background as they blended together and became indistinguishable without it. Since these changes have only affected the background and menu bars, these changes are to be implemented for every screen, not just the Media Status screen.



### Logs screen

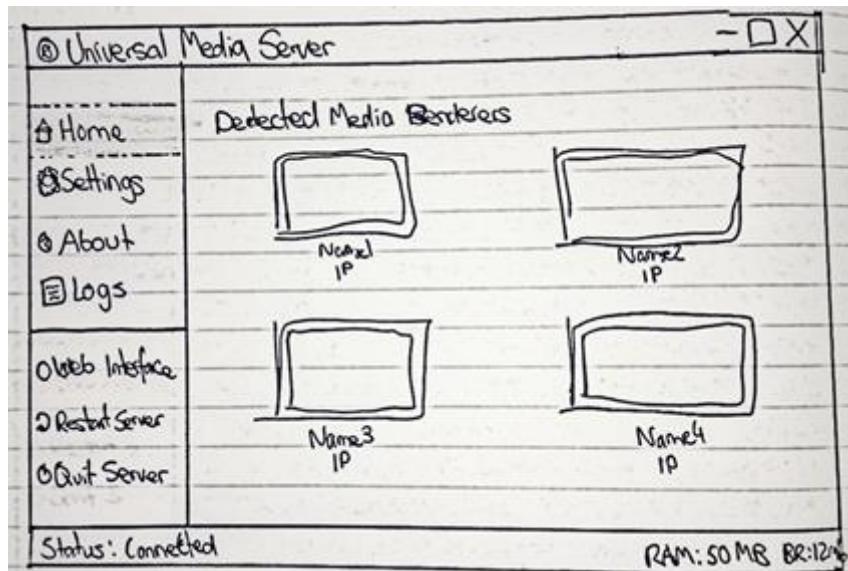
The logs screen has gone through no changes between the low fidelity and high fidelity prototypes, aside from the program wide changes mentioned in the Media Status screen.

## 6.2 Lachlan Mackenzie

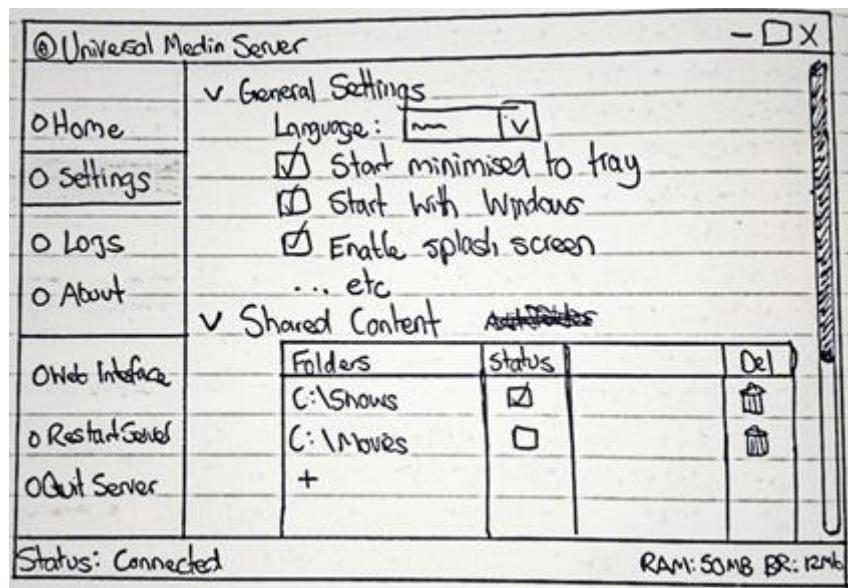
Contributor(s): Lachlan

### 6.2.1 Low-fidelity Prototype

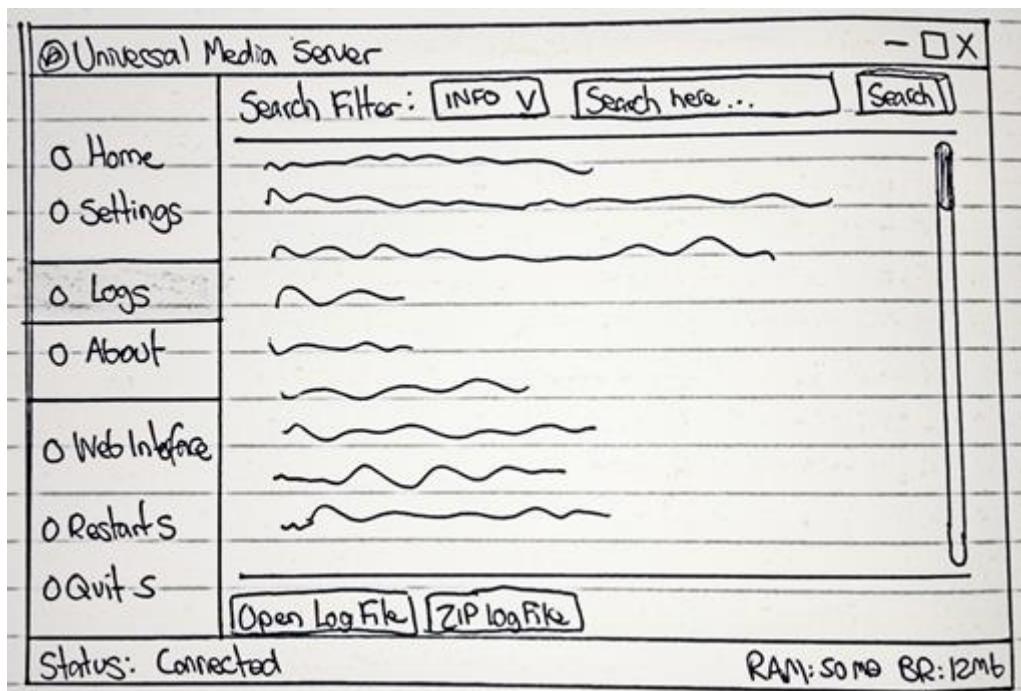
This low-fidelity prototype does not show every possible screen. For a detailed explanation on the design see [Section 6.2.2](#) below. Otherwise the explanation would be repeated as very little changed between the low-fidelity and high-fidelity prototype.



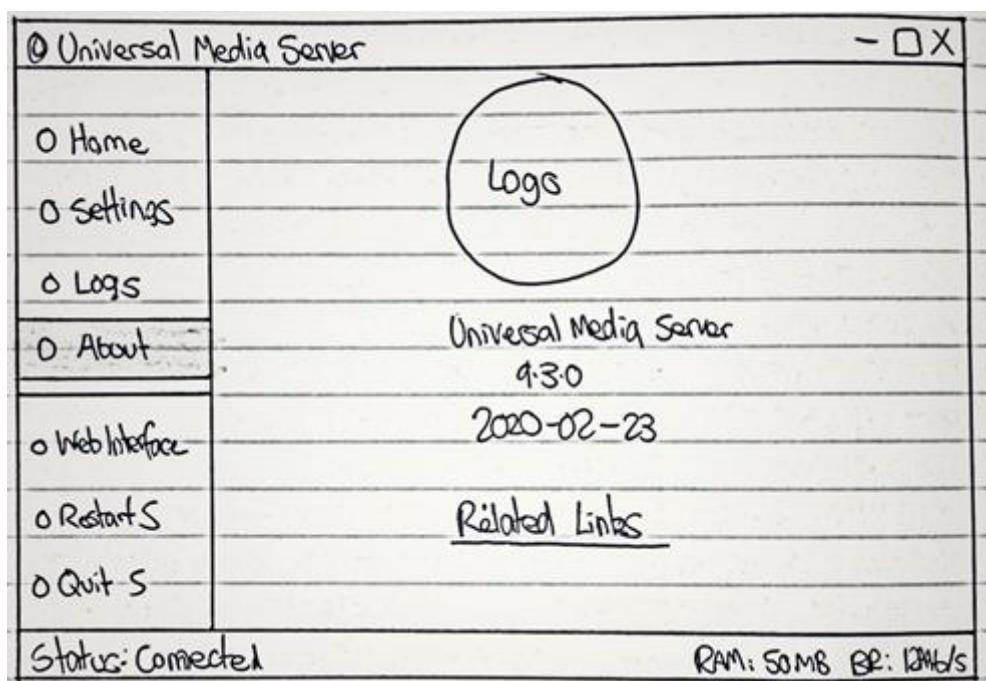
Home Screen to greet the user



Settings Screen



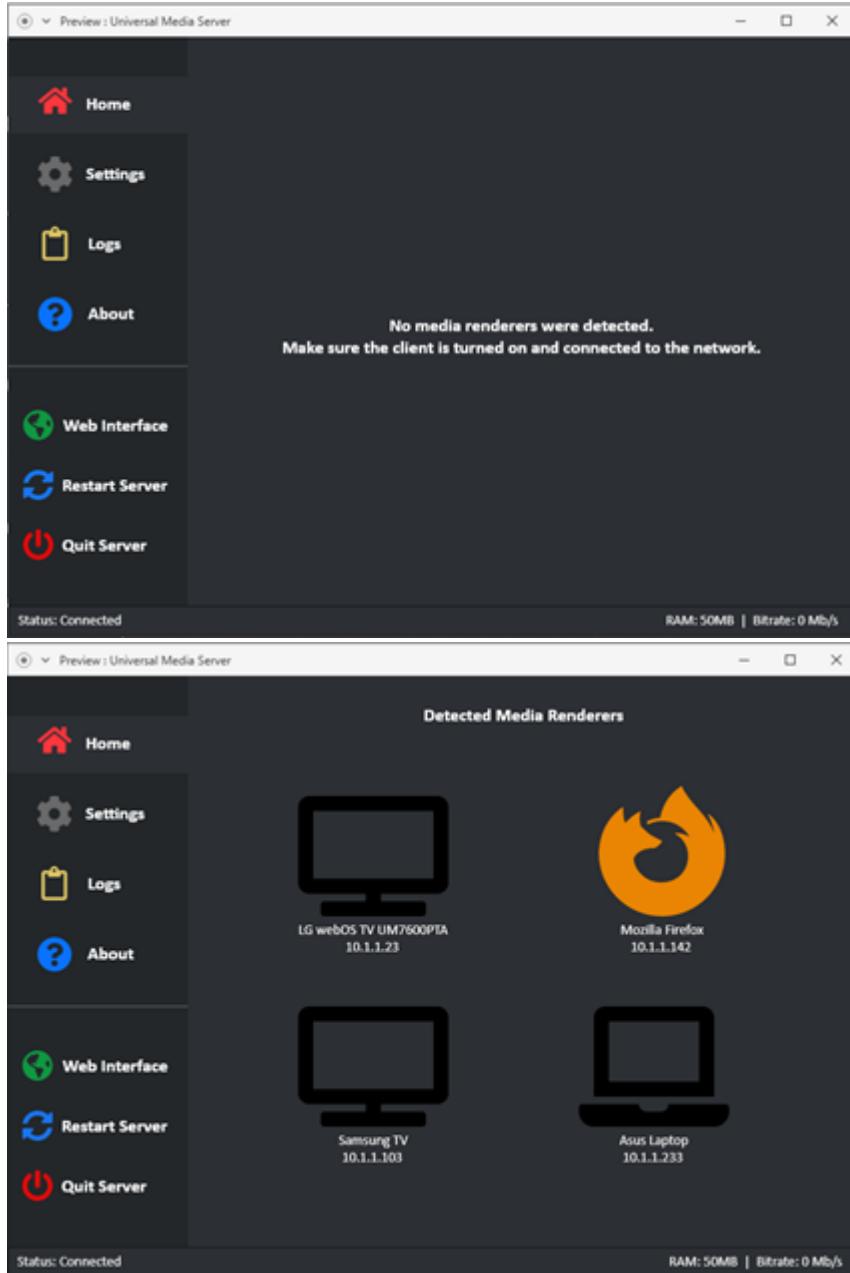
Logs screen



About Screen

## 6.2.2 High-fidelity Prototype

### Home Screen



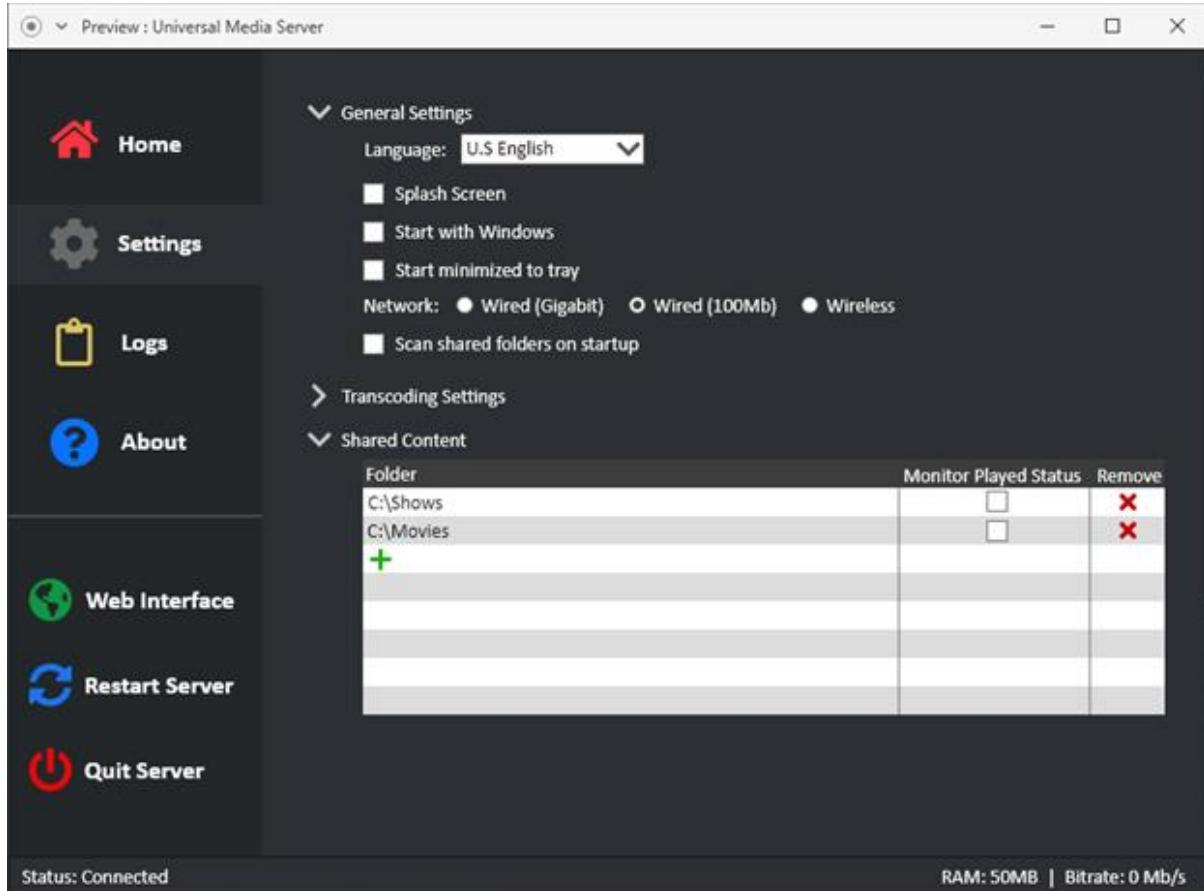
This is the screen that will greet the user once the application has started.

On the left is the main navigation pane. The upper section is for the user to navigate through the different screens of the application. The lowest section contains functions for the media server that is run when the application starts. The 'Web Interface' button will launch a front end to the server in your default web browser. The 'Restart Server' will restart the server and change colour to let the user know when it has successfully restarted. The 'Quit Server' will be used to shut down the media server with a confirmation dialog. All of these buttons in the left pane will have tooltips appear when they are hovered over to give the user a better idea of what they're used for.

At the bottom is a status bar for notifying the user the status of the media server and how much memory and bandwidth is currently being used. This status bar is visible no matter where you are in the program.

The ‘Home’ tab will show all available devices that can be streamed to. If no streaming devices are connected to the network, a simple message will be displayed explaining why and what you can do to fix it. If there are devices detected, the application will try to lay them out dynamically in a grid like fashion as per the current implementation.

## Settings Screen

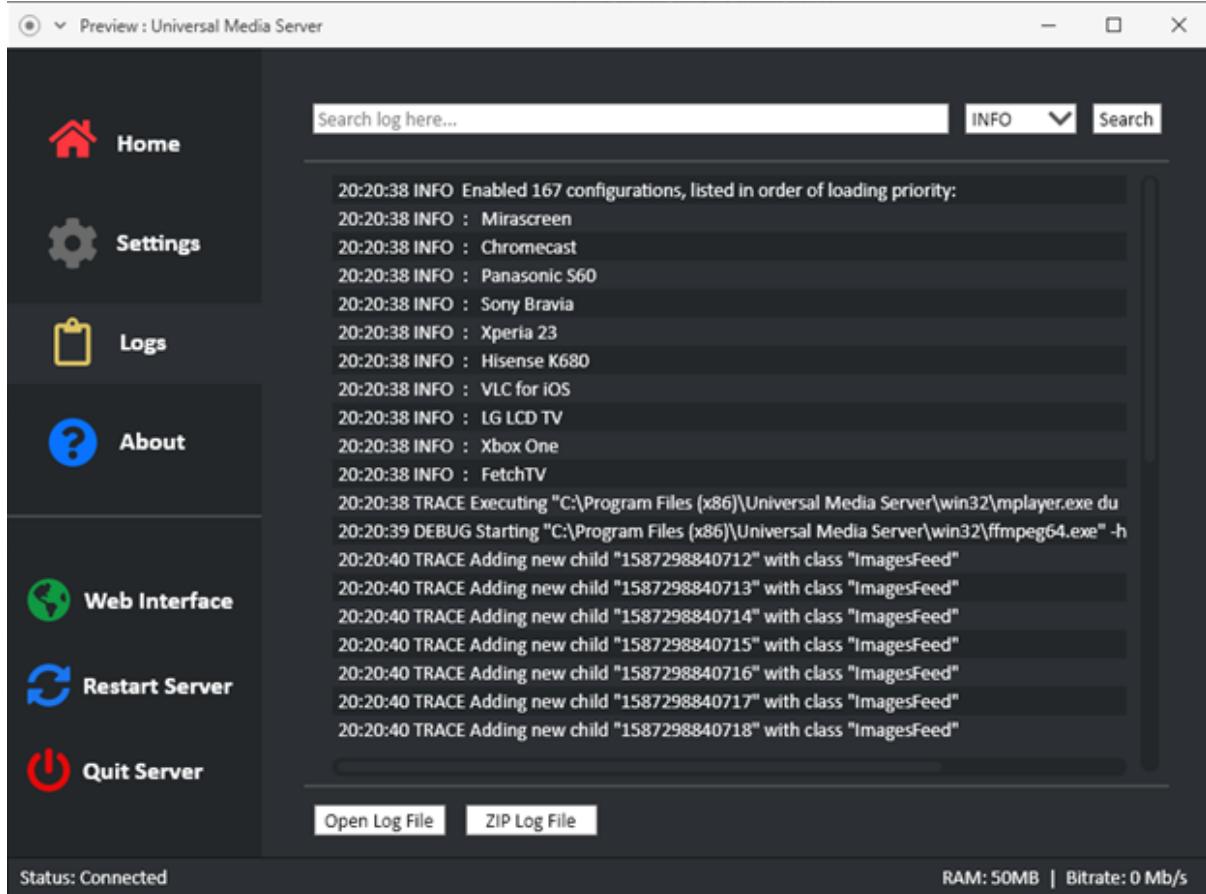


This is the screen for changing the applications settings. Each settings section can be collapsed or un-collapsed using the chevrons/arrows next to the section heading. This can help the user reduce the amount of clutter on their screen.

The Shared Content setting is used to add folders containing media that the user wants to stream to other devices. This table shows all currently shared folders, they can be removed from the table at any time by using the red cross. The checkbox is used to tell the application that it should track whether the media has already been played so the user can keep track of what they have watched.

The green plus icon is used to add another folder to the shared folder list, this button will bring up the operating system's folder selector. If the table gets too long, a scroll bar will appear to the right of the table.

## Log Screen



This screen is for showing the log file of the application to the user in a presentable and searchable manner. The user can search the log file by keywords and a filter for the logging level (e.g. INFO, DEBUG), the search will then filter the log for the keywords being searched for.

The two buttons down the bottom of the Logs screen are to help with the debugging process for a common user. The 'Open Log File' will open the entire log file which stores all logs since the application has been installed on the user's system. The 'ZIP Log File' will compress the log files into a zip file that can be easily sent to the developer for debugging when the user is experiencing a problem.

## About Screen



The About screen is for displaying the version of UMS that is currently installed and running on the user's system. There is also a link to the documentation/guide for helping users get comfortable with the application.

There are also many related links displayed on this tab as they could be helpful to the user who might want to learn more about media players and servers. All of the links are not displayed in the picture above.

## Usability Goals Achieved

### Efficiency

The program is very efficient for performing many different tasks. This is due to the simplicity of the design, somewhat complex tasks for this program can be completed by the user with relatively few mouse clicks.

### Persistence

Whenever the user makes a modification to their settings in the program whether it be a simple checkbox or another entry in the shared content table the program will save this to a file. This makes sure the user won't lose any of their settings when coming back later to the program.

### Safety

The program protects the user from performing unwanted actions that may be triggered on accident. Confirmation dialogs are used to make sure the user really wants to remove a folder from the shared content or close the server.

### **User Experience Goals Achieved**

#### Pleasurable

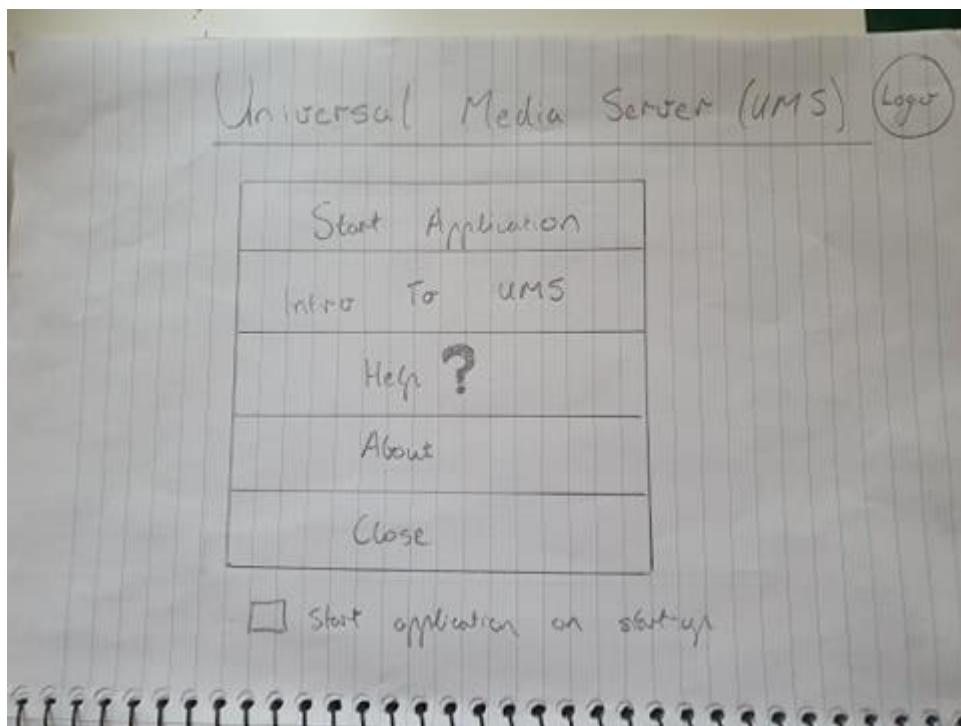
The colour theme chosen is appealing to the eye at both daytime and night-time. The colourful icons really pop against the dark theme and give the program a vibrant feel. The program is modern, clean and is pleasurable for the average user to use.

## 6.3 Alex McLeod

Contributor(s): Alex

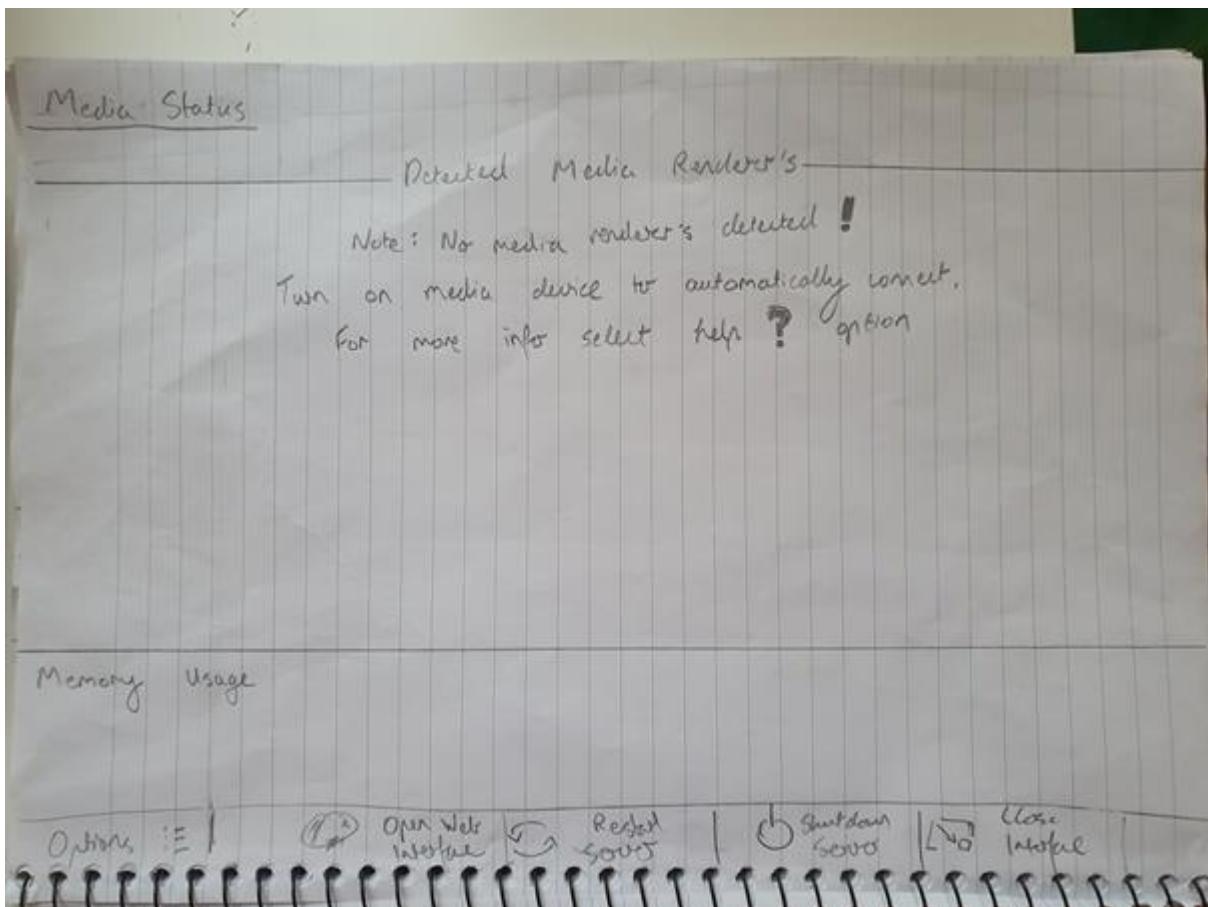
### 6.3.1 Low-fidelity Prototype

Start Menu:



This is the first basic design of the start menu. Because this is the first menu the user sees I tried to make it as clean and simple as possible with the title of the application along with a logo and a few buttons with large text to make them easy to read for the user. The main reason for creating this screen was to provide new users to the program an easy way to find help on how to use the program and information on the program. The first button I included was one to start the actual application. The second a button for launching an introduction video for the application to provide help to new users of the program. The fourth a help button which will open a window explaining how different parts of the application work in detail. The fifth an about button for information on the application such as development, support etc. Lastly a button for closing the application. I also included a checkbox at the bottom of the screen which when ticked would cause the application to start from the Media status screen instead of the start menu for convenience.

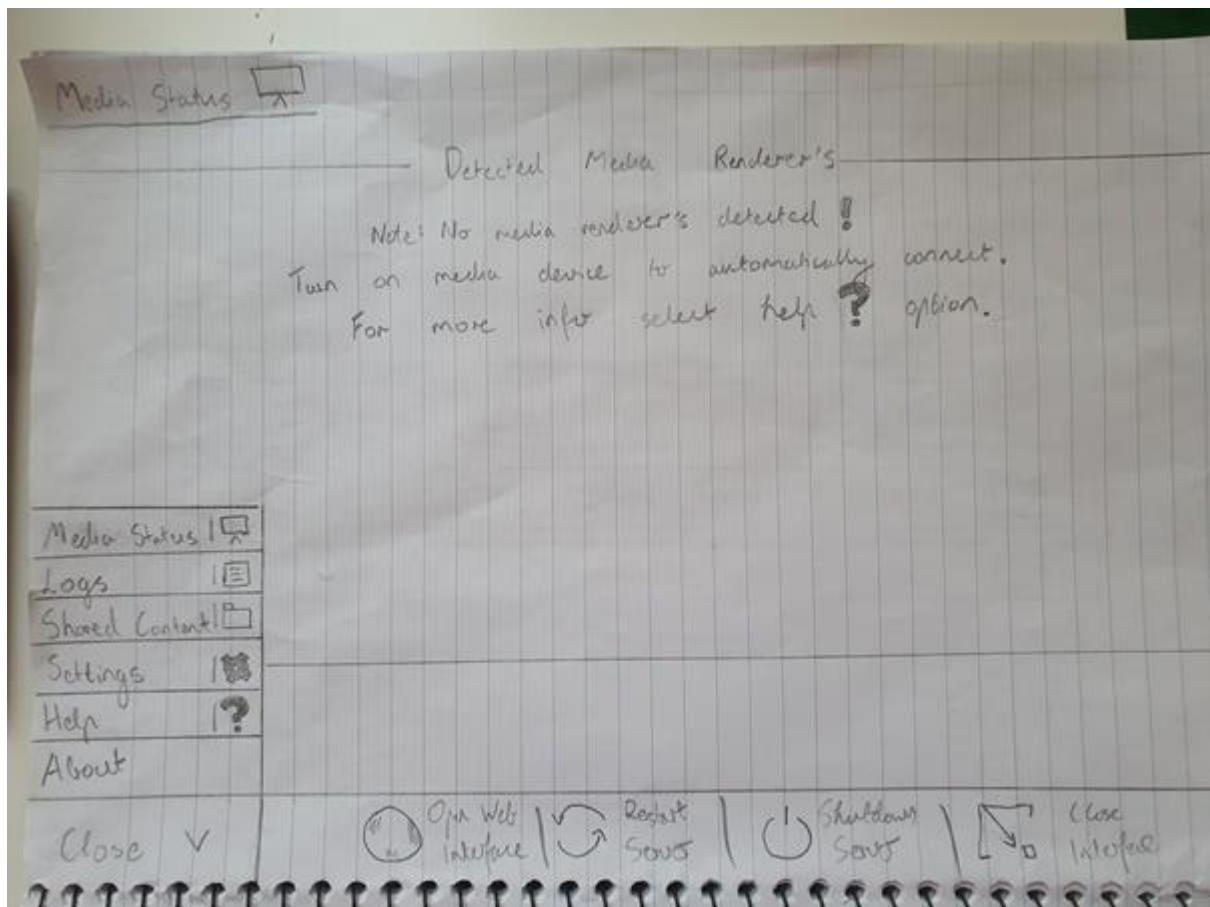
Media Status Screen:



This is the first design of the Media Status Screen which starts after the user selects start application from the start menu. This screen displays the different devices which have been automatically detected by the program that can be used to stream different media. I chose to make my application as clean as possible with minimal clutter by placing the name of the screen at the top left corner, giving a large space for different devices to be spread out and by placing all the important buttons for the server and the application at the bottom of the screen. These main buttons are at the bottom of each screen of the application giving the user easy access to critical options wherever they are in the application. These buttons also have their own respective icons related to their function which should be familiar to most users allowing them to easily understand and distinguish their uses. In terms of icons I chose to include a list of lines for the options button, an earth symbol for the web interface button, two arrows in a cycle for the restart server button, a power symbol for the shutdown server button and a minimize symbol for the close interface button. These icons add to the visual appeal of the screen and allow the buttons to be easily distinguished.

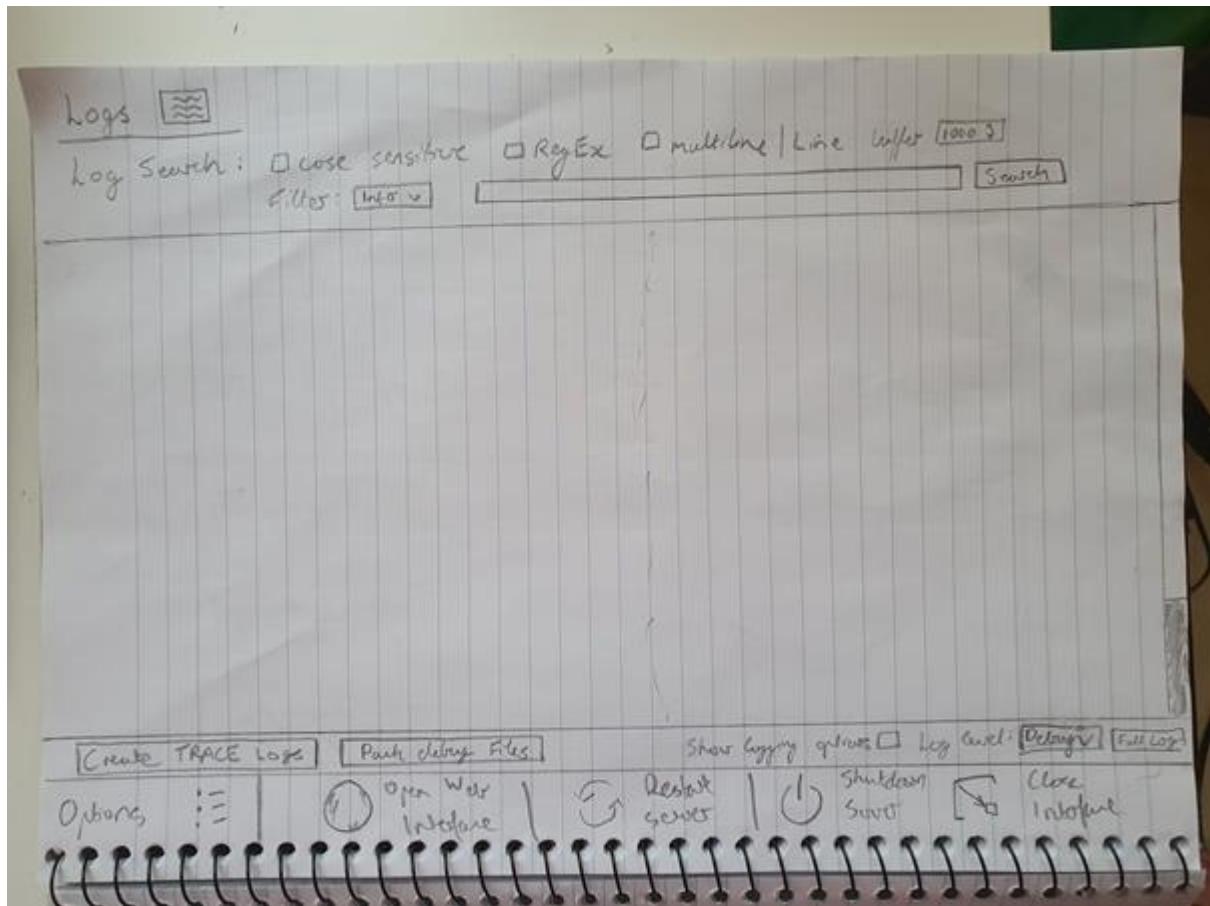
When no devices are detected I chose to display text stating that none have been detected and how they can get their devices connected providing more help to the user. The memory usage is displayed at the bottom of the detected media renderers which shows the memory usage of the server as more devices are connected.

Media Status Screen (Options Button Selected):



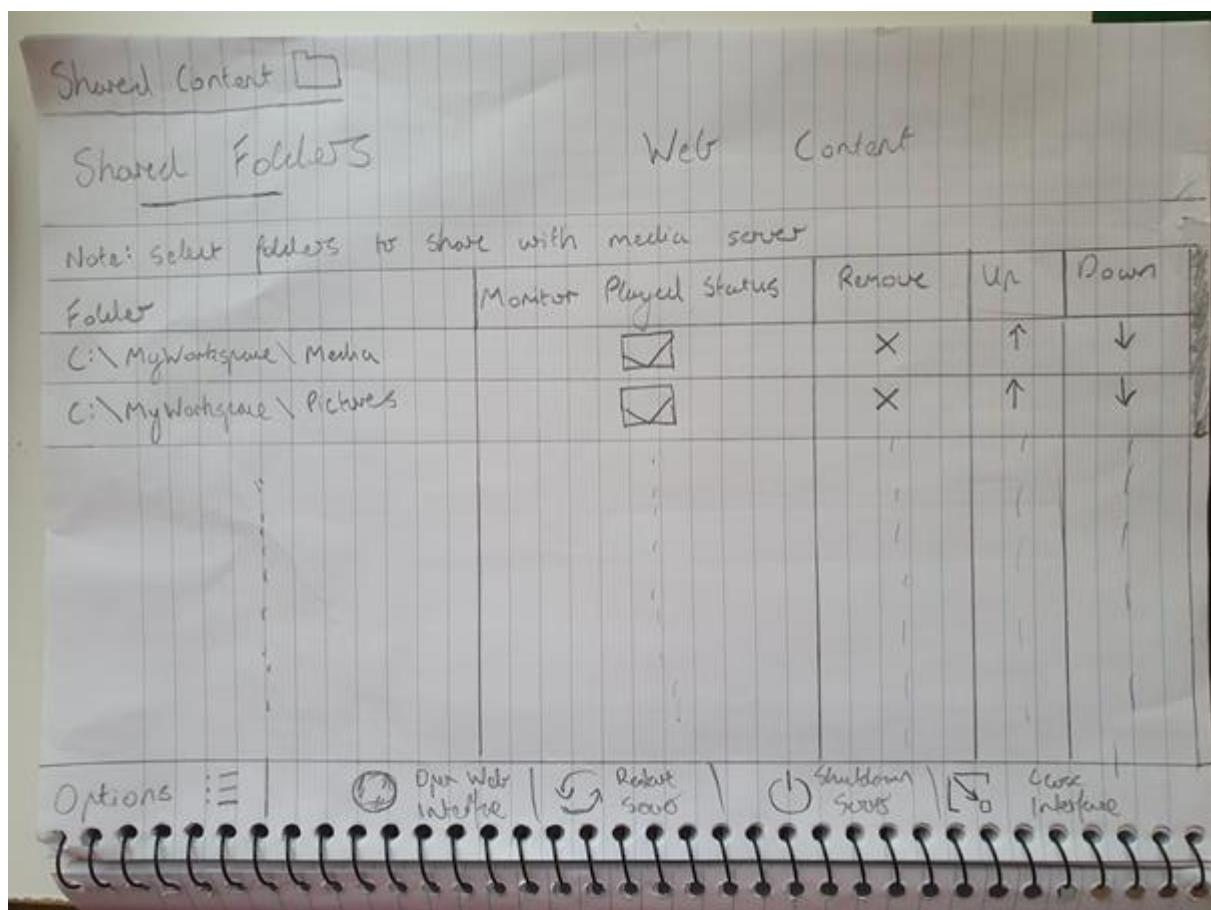
This image represents what will happen when the options button is pressed. As we can see it displays the different screen options available within the application through a combo box. It acts as a simple way of navigating through the application. I chose to use this as the main way of traversing the application as the expanding menu reduces the amount of room taken up by the different menu options making each screen cleaner with less unnecessary information being displayed when it doesn't need to be. I also included symbols for each option that should be familiar to most users allowing them to more easily identify the purpose of each option.

## Logs Screen:



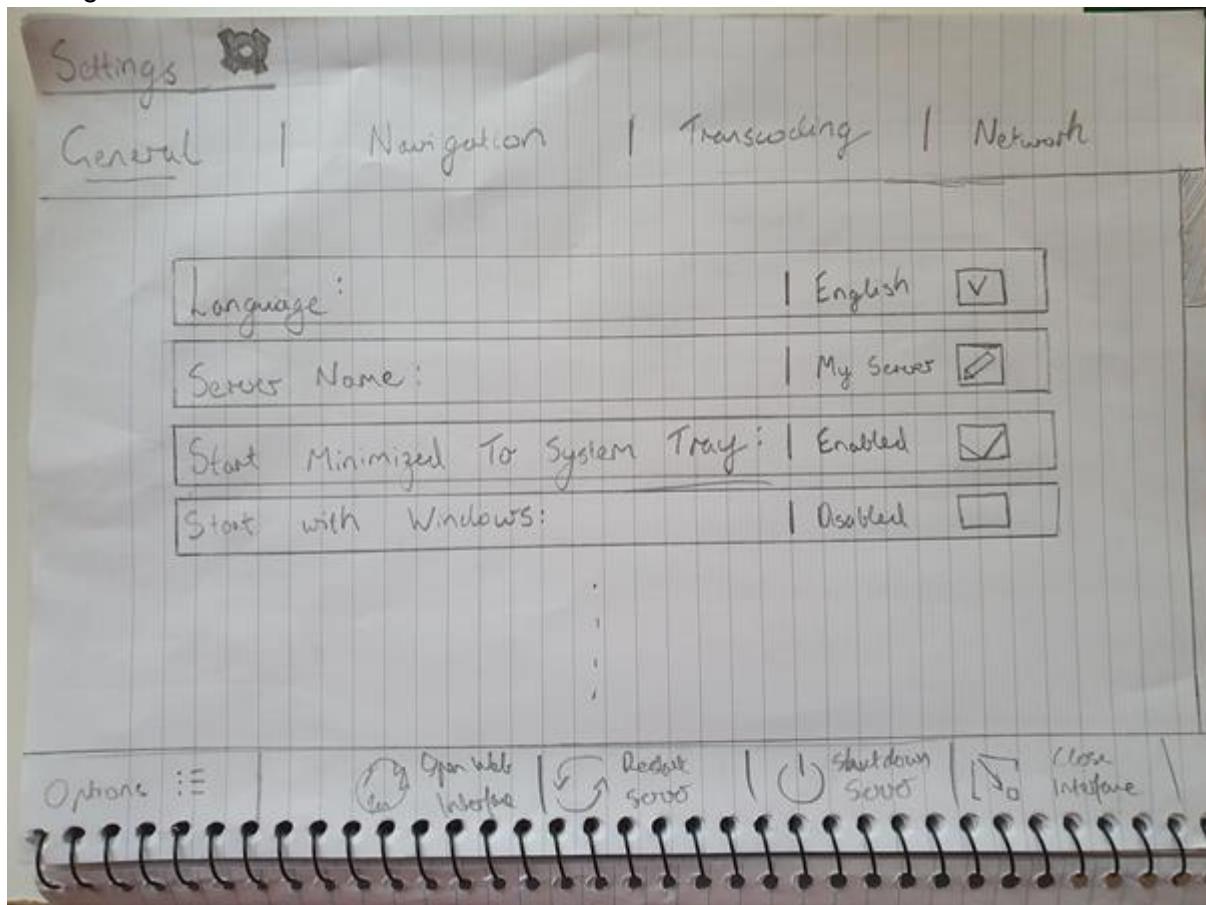
This is the first design for the Logs screen. The main purpose of this screen is to display the logs created by the server. Allowing the user of the application to understand how their media server is running. If there are any issues it allows them to find them and potentially troubleshoot the issue. Similar to the media status screen it displays the title at the top left with the log options in the middle and the main application buttons at the bottom. I chose to make the layout of each screen quite similar allowing the user to easily find information and quickly become familiar with the application. The middle of this screen contains a scrollable pane containing the different log information and allowing the user to scroll through the different log information. Above this pane are different options allowing the user to customize their search preferences when trying to find information in the log pane. Below this pane are more advanced options for creating trace logs, packing debug files, etc.

Shared Content Screen:



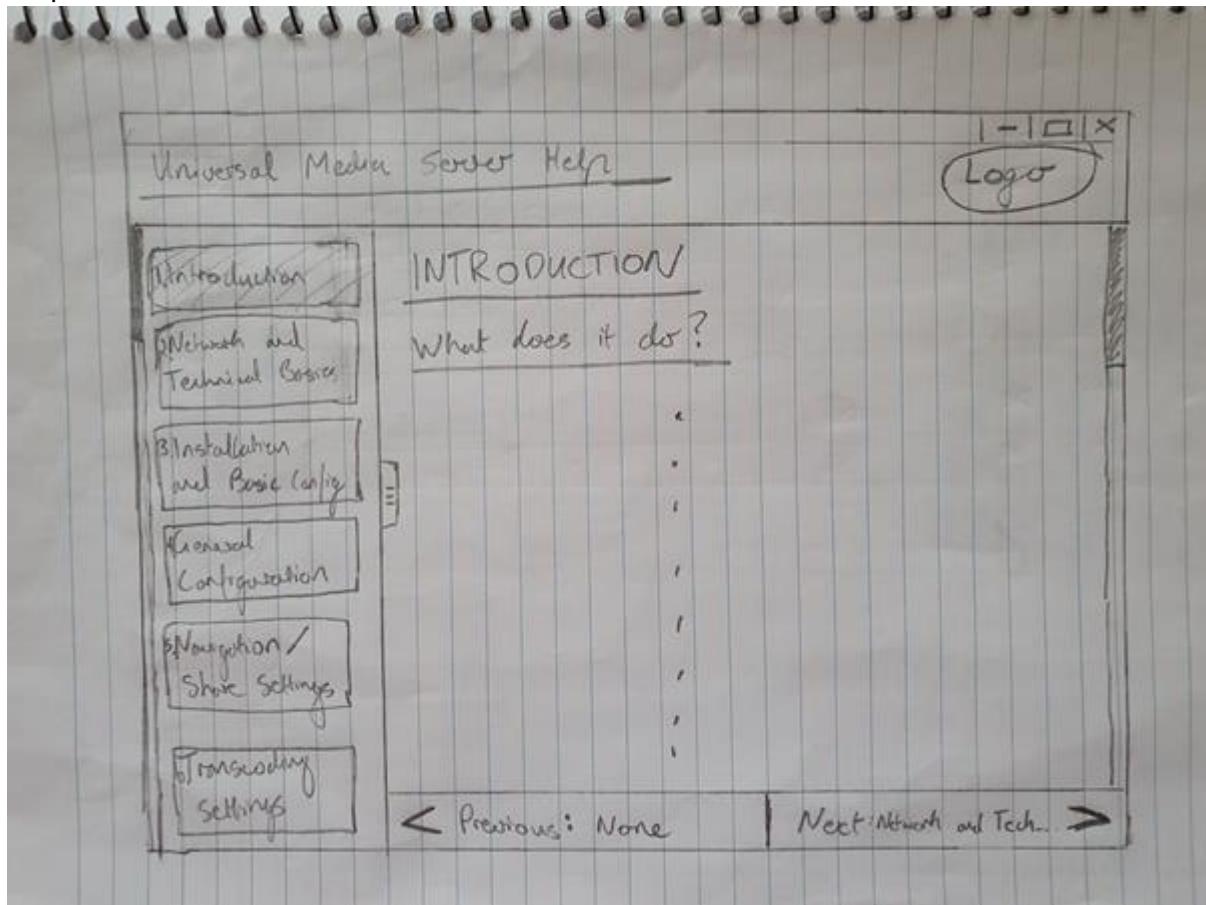
This is the first design of the Shared Content Screen allowing the user to select the different folders and web content that they would like to share with their media server. I chose to split this menu into two tabs one for sharing folders from their computer and another for sharing web content. I chose to split it this way so that the user can easily distinguish the media they are sharing and so there is less information crammed onto one screen. Each tab containing a table displaying the folders or web content they are sharing. Along with the name of what is being shared in one column there are also additional columns to easily monitor played status through a checkbox, remove the item via a button and move the order of the folder up and down via buttons. Each of these buttons having a relevant symbol which should be familiar to most users, so they can easily customize each entry.

Settings Screen:



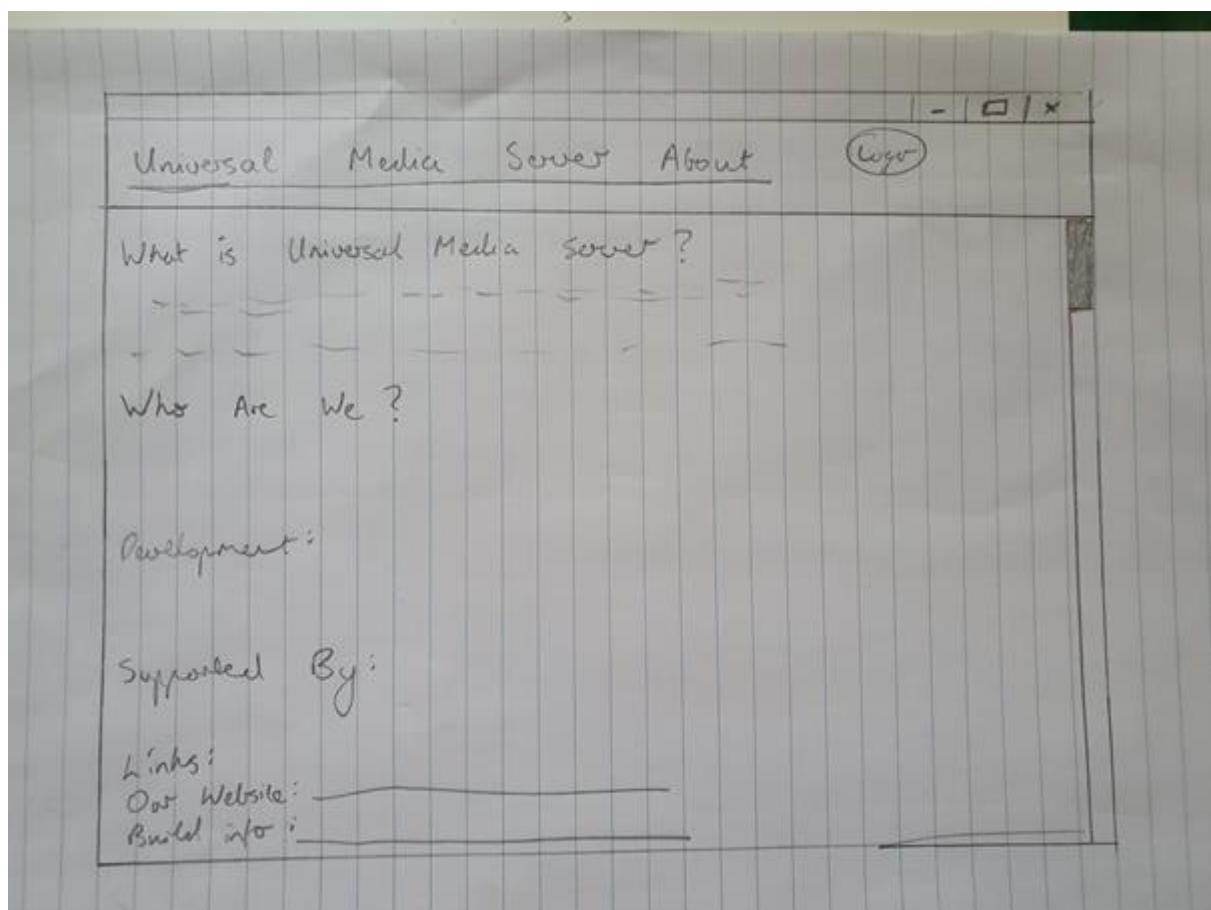
This is the first design for the settings screen. It allows the user to customize different settings related to the media server. Similar to the shared content screen I decided to organize the different types of options via tabs. Each of these tabs being related to distinct setting areas of the application so that the user can easily locate particular settings. These being either General, Navigation, Transcoding or Network settings. Each tab contains its own scrollable pane allowing the user to easily browse the different options available. Each option is contained within a label and designed in a similar way with the description of the option on the left and the user input for the option on the right (e.g. a text field). This consistent formatting gives the settings menu a clean look.

Help Screen:



This is the first design for the help screen. This screen is a separate window to the rest of the application which is opened when the user selects the help option from the expandable menu. I chose to have the help screen in a separate window so that the user can view the help screen at the same time they are using the application making it more convenient than having the user switch back and forth each time they want to view the help information. I laid out this screen with the title of the screen at the top and below this the actual help information. On the left of the screen is a scrollable menu. This scrollable menu contains buttons for each of the different help topics. When one of these buttons is pressed the information for that topic is displayed on the right in a scrollable pane of its own. I chose to use a side scroll menu because there is a lot of help information so making it easy for the user to navigate and search for a particular topic is important. This menu hence makes the help screen more convenient allowing the user to find the main topic they are looking for and instantly display it on the right rather than having to scroll through hundreds of lines. At the bottom of the Screen is two buttons so that when the user has finished a particular section, they are able to quickly switch to the previous or next topic.

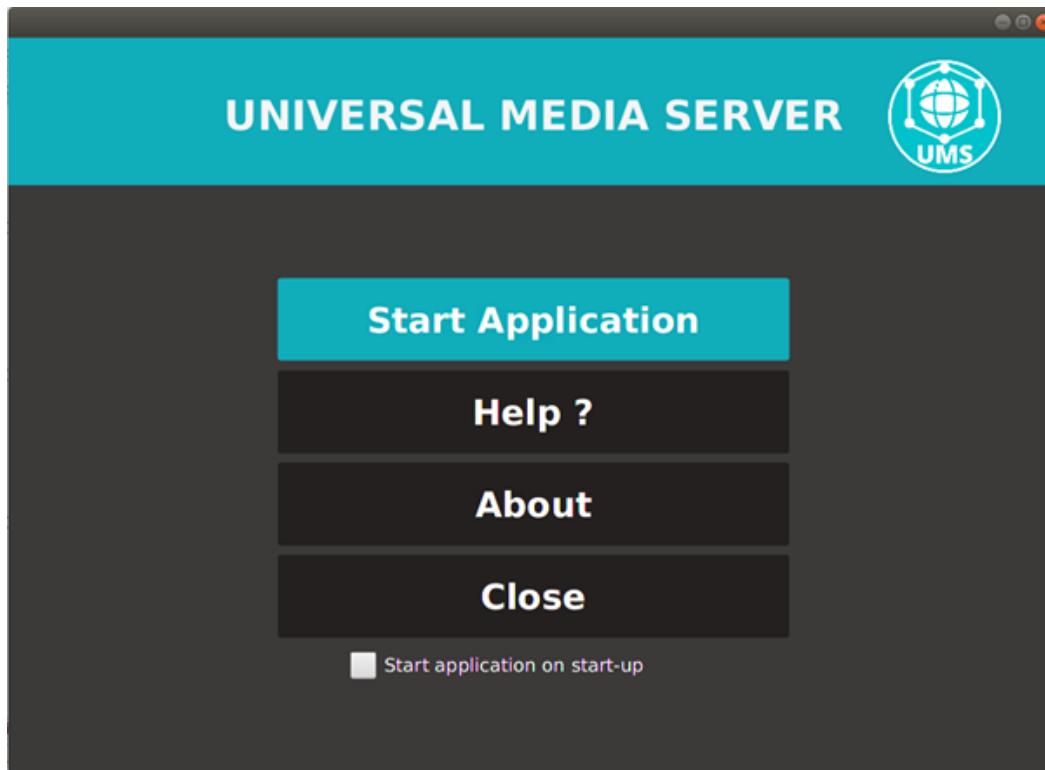
About Screen:



This is the first design for the about screen. This screen is very simple as it doesn't contain a large amount of information. Similar to the help screen it has its own separate window for convenience. This menu simply contains a title at the top and below it a scrollable pane so the user can easily scroll through information to do with development, support, and external links.

### 6.3.2 High-fidelity Prototype

Start Menu:

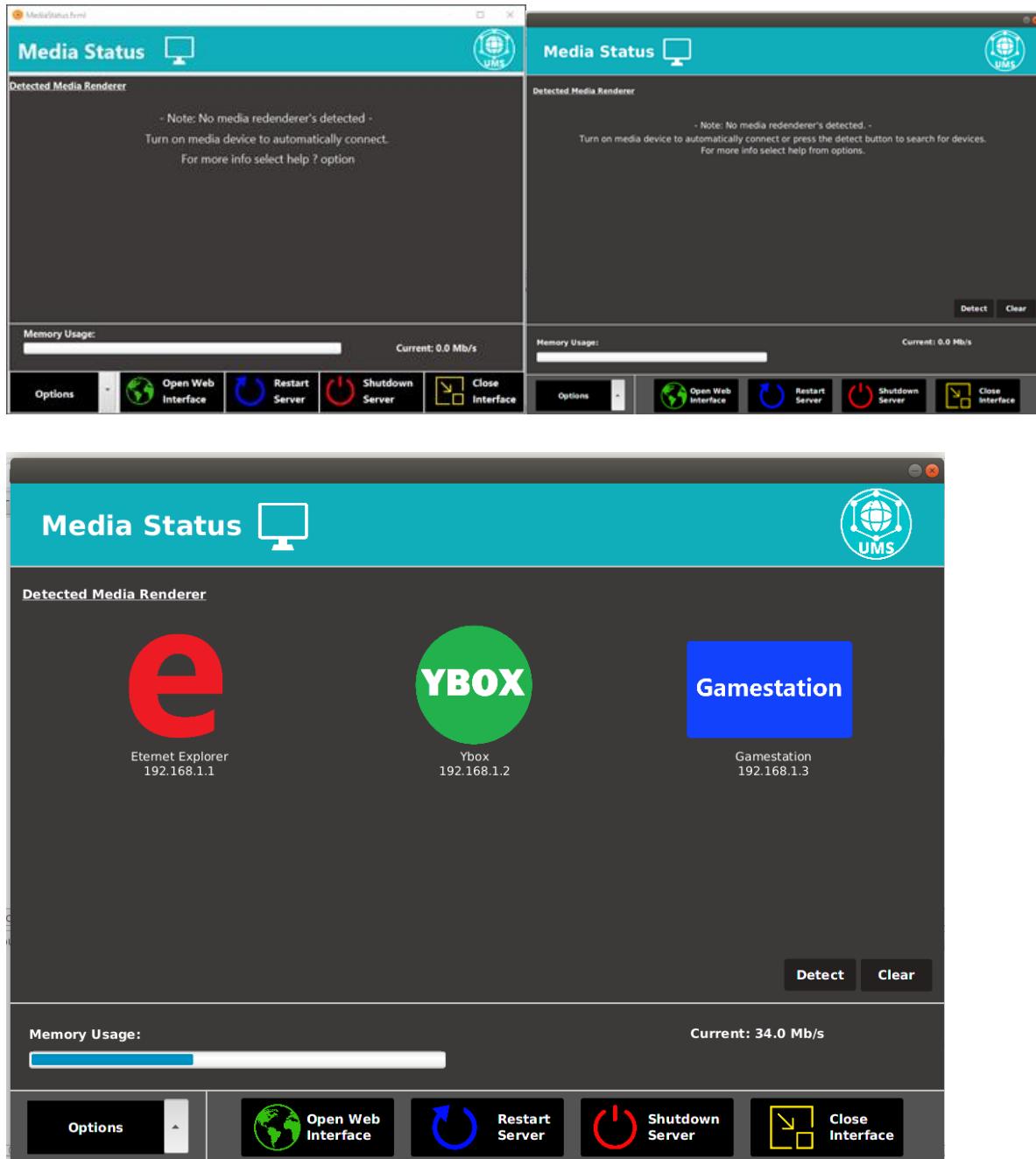


This is the high fidelity design of the first start menu of the application. It is an upgrade of the low fidelity design in many ways. To begin with I have included colour to create visual appeal and emphasise parts of the application. It took me a long time to discover a colour scheme that I liked but eventually I chose to use white and aqua against a dark grey colour. I chose this scheme because by using a contrast of colours I was able to highlight parts of the screen allowing the user to easily see and notice parts of the screen such as buttons more easily. For example, I used the white colour for the text in the buttons and a dark grey colour for the background causing the text to stand out. At the top of the screen the bright aqua colour causes the title of the application to pop out to the user and catch their eye, which may in turn cause them to remember the name of the application. For each of the buttons I also designed them so that they change colour when the cursor is hovered over them creating user feedback for the user and making it a more satisfying user experience.

At the top right of the screen I also created my own logo for the application that matches the colour scheme of the application. I chose a simple design for the logo with the different symbols representing different functions of the application. The symbol in the middle that is a sphere represents the universal nature of the application and the connected dots on the outside represent the ability of the application to connect to different devices for streaming. This simple design was chosen so that it would match the clean aesthetic of the rest of the application.

I also chose to remove the button "intro to ums" from the low-fidelity design as I decided that it was redundant and not worth the time creating when there is a help button already present that explains every part of the application.

### Media Status Screen:



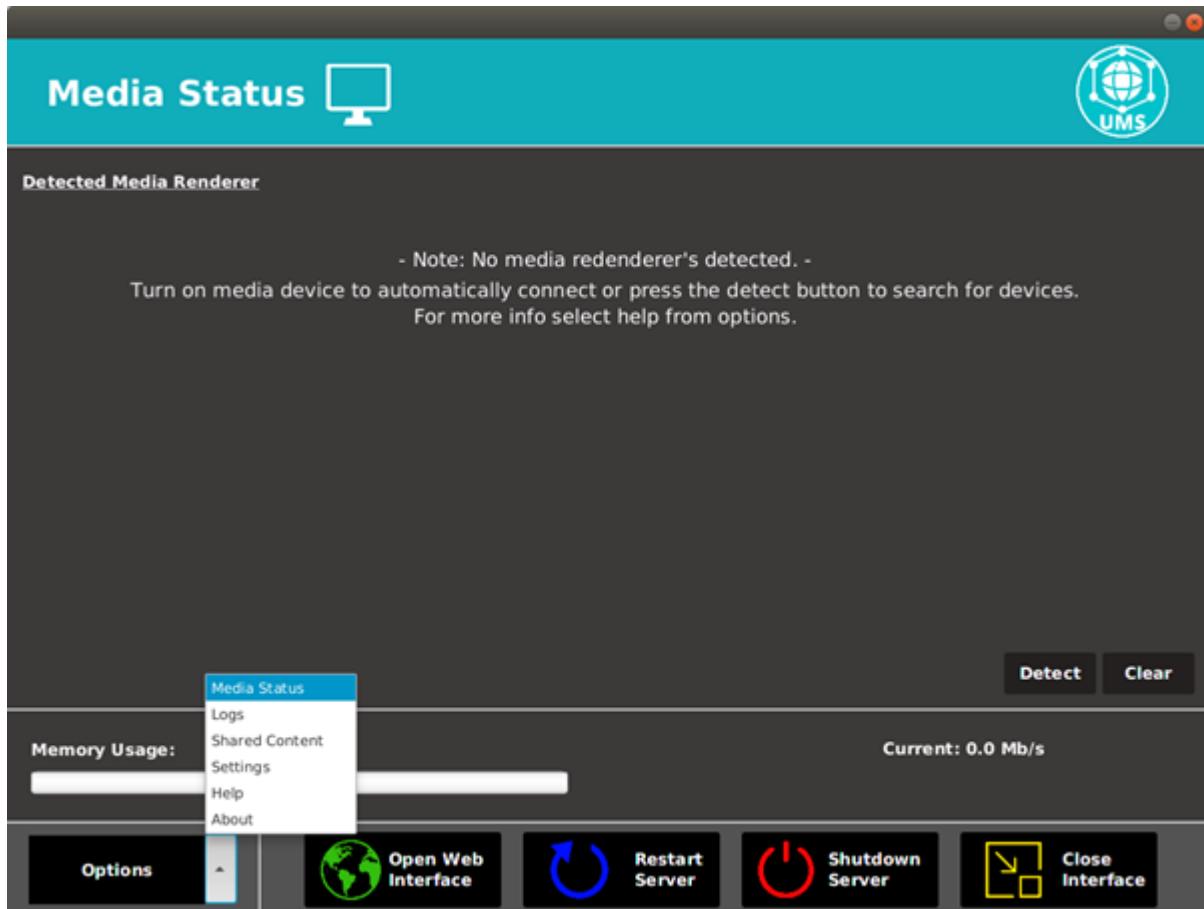
This is the high fidelity design of the Media Status Screen. On the top left is the first high fidelity design, on the top right is the final high fidelity design and at the bottom is the final high fidelity design with media renderer's detected. Starting with the first high fidelity design I improved the initial low-fidelity design by adding colour and more icons. The colours I chose to include were based on the colour scheme of the rest of the application creating consistency amongst the screens and making it visually appealing to the user. At the top of the screen I chose to use an aqua colour to make the title of the screen pop out to the user. At the top of the screen I also included a computer monitor icon. This icon relates to the main purpose of the screen which is to detect media devices. By including this icon, it allows the user to easily distinguish the different screens of the application. At the bottom of the

screen I added colour to the icons of the main buttons of the application so they can be easily distinguished. As well as this I also added a feature that causes the buttons to turn an aqua colour when hovered over creating user feedback.

With the second image I re-adjusted the layout and design of the first high fidelity design. After receiving feedback from different sources, I realised ways of improving the layout and overall look of the interface to make it more appealing to the user. At the bottom of the interface I chose to make each of the buttons slightly smaller and more spread out so the bottom of the screen wouldn't look too cramped and the buttons could be easily distinguished. I also chose to reduce the width of the dividers splitting up each pane and removed the white border on the outside of the screen for a cleaner look. As well as these changes to the look of the screen I also added tooltips to the buttons at the bottom giving them better explanations for what they do for the user. I also added a detect button and clear button for detecting devices manually and clearing any connected devices, allowing more functionality. These buttons also have tooltips attached to them to give further explanations for what their uses are.

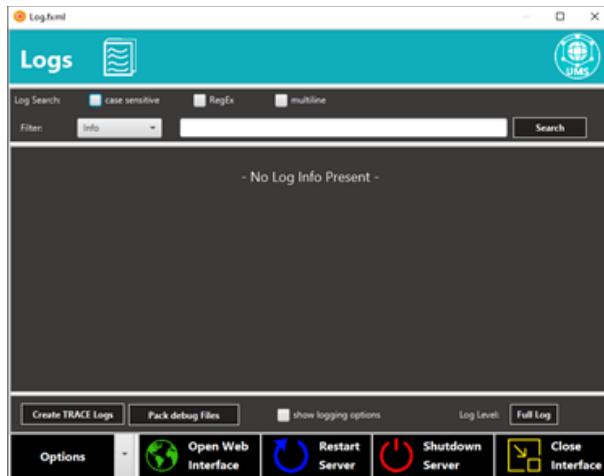
The bottom image is the final high-fidelity design that shows what the screen looks like when devices are detected. Through this image it emphasises why I made the background a dark colour. By doing this it causes these different devices to pop-out to the user grabbing their attention. It also allows us to easily see the white text. This screen also demonstrates how at the bottom of the screen the memory usage bar increases as devices are added and the mb/s increases.

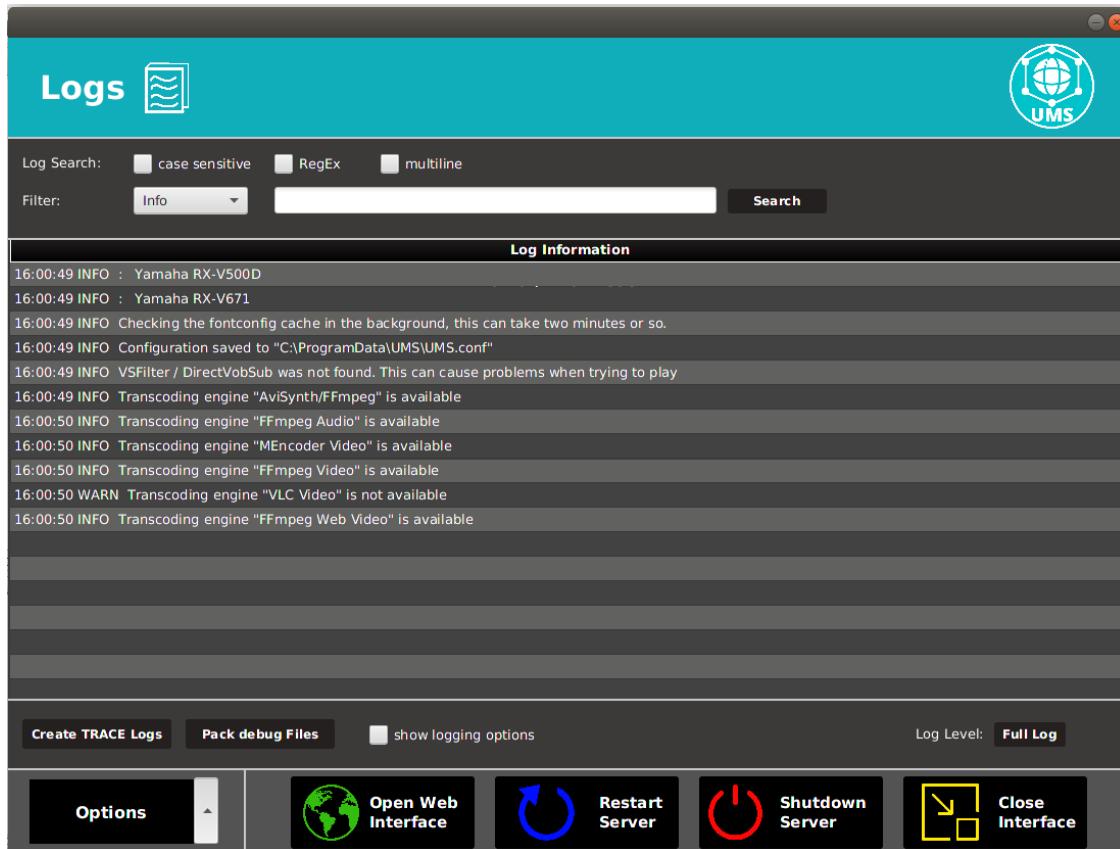
Media Status Screen (Options Button Selected):



This is the final high fidelity design of the media status screen with the option button pressed. This design has changed significantly. It has become less complex and just includes the name of each screen. This is mainly because of the limitations of NetBeans which did not enable me to add any icons within the combobox itself. Hence, I just decided to have a clean look for the options with just the words. These options also provide user feedback as they become highlighted when hovered over.

Logs Screen:





This is the final high fidelity design of the logs screen. At the top is the first high fidelity design and on the bottom is the final high-fidelity design. This design in general is quite similar to the low fidelity design with a number of improvements. Similar to the other high-fidelity designs I included colour to add to the visual appeal of the application. I also included the final icon for the screen which is a book. I thought this was a suitable icon because like a book log files record information of events that have occurred. It also gives the screen a bit more visual appeal which is needed on a screen where it mostly contains just lines of text. The logo included also adds to this visual appeal. Similar to the other screens each button includes a hover effect where the button turns an aqua colour when the mouse is moved over it adding to the user feedback.

From the first high fidelity to the final high-fidelity design I made slight improvements in the look of the interface. This involved separating the buttons at the bottom, reducing the width of the dividers between panes and removing the white borders around buttons resulting in a cleaner look. I also included the table containing log information. I styled this table to match the colour scheme of the rest of the screen and added alternating colours for each row so the user can easily distinguish between different rows.

## Shared Content:

Folder Name	Monitor Played Status
myFolder	<input type="checkbox"/> monitor
myCoolFolder	<input type="checkbox"/> monitor
myGreatFolder	<input type="checkbox"/> monitor
myCrazyFolder	<input checked="" type="checkbox"/> monitor

Shared Folders Web Content

Note: Select Folders To Share with Media Servers

Add Folder Remove Folder Scan Folders to Add to Server

Options Open Web Interface Restart Server Shutdown Server Close Interface

Folder Name	Monitor Played Status
myFolder	<input type="checkbox"/> monitor
myCoolFolder	<input type="checkbox"/> monitor
myGreatFolder	<input type="checkbox"/> monitor
myCrazyFolder	<input checked="" type="checkbox"/> monitor

Add Folder Remove Folder Scan Folders to Add to Server

Options Open Web Interface Restart Server Shutdown Server Close Interface

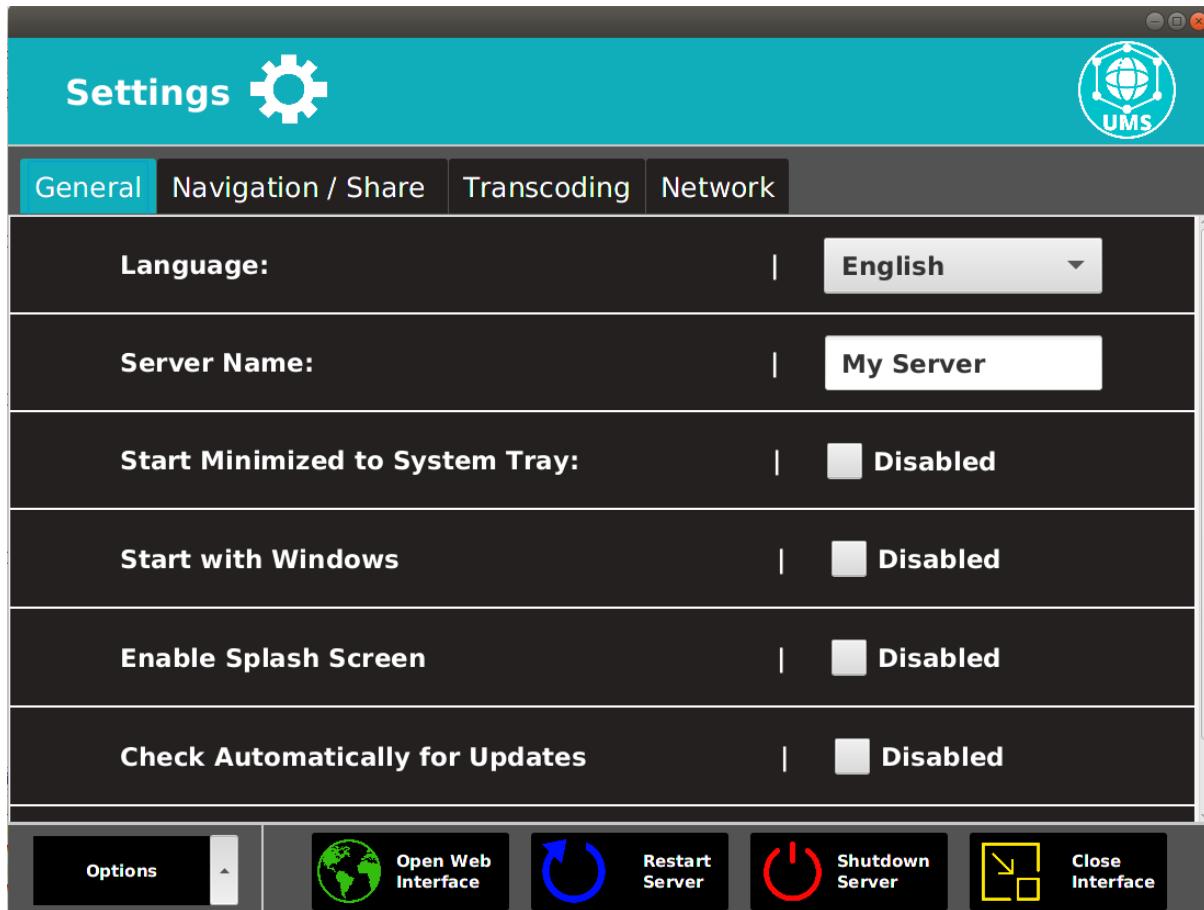
These are the high-fidelity images of the shared content screen. The top image is the first high fidelity image I created, and the bottom image is the final high fidelity image after review.

Starting with the first high fidelity image from the low-fidelity design the same changes were made in terms of colour and icons as the other high fidelity screen. On this screen the icon I chose to represent shared content was a folder. This being because folders hold content and most people relate the symbol of a folder to file sharing. Hence it allows the user to easily understand what this screen is used for and it increases visual appeal. From the low-fidelity design I also changed the layout of the table. I started by removing the ability to move the shared folders up and down as it seemed like a redundant feature when you can sort the folders anyway by clicking the top of the column. Rather than having the remove button in the table I placed it below the table allowing a lot of room to be saved. I also added an add folder button where when pressed the user can select a folder from the file explorer to share. The scan button has also been added to the bottom to allow users to scan the shared

folders. When tab is selected it glows aqua ensuring the user knows the current tab they are on. When the web content tab is selected it also displays a table. This table is slightly different as it displays the type of media, folder name and the source address of the content. It also has an add button allowing a user to add web content to be shared and a remove button to remove any web content.

For the final design at the bottom I changed the look of the table making it match the colour scheme of the screen to give it a better overall look. I also added a tooltip to the scan folders button in the case that the user is wondering which folders are being scanned.

Settings Screen:



This is the final high fidelity design of the settings screen. From the low-fidelity design there are similar changes as the other screens including the added colour for a better look and the added icons. The cog icon is a universal symbol for settings on most applications. Hence using it on this settings screen is appropriate as it instantly allows the user to recognise this screen as the settings screen. I also increased the size of each label making the text of each setting option larger increasing user friendliness. Pressing any of the tabs at the top of the screen will change the pane displayed below it with settings relevant to the option selected.

Help Screen:

Help Menu

# Universal Media Server Help

## ?

### 1. Introduction

Universal Media Server (or UMS for short) is a program that streams media like video, audio or images from your computer or the web to a device called a "renderer" that can be anything that supports DLNA like a game console, a TV, a computer software or a mobile phone.

This might sound simple, but it is in fact a complex matter. There are dozens of media formats in the world, and your console or TV only understands a couple of them. UMS allows you to browse media on your favorite renderer without having to worry about the details of what you are viewing. UMS takes care of retrieving the media (from file, network or web) and translating it to the best format that your device understands.

### Media Renderers

UMS is a DLNA compliant UPnP Media Server. It was originally written to support the PlayStation 3. However, other renderers support the same standards as the PlayStation 3 and UMS has been expanded to support a range of them, including Xbox 360 and various Samsung and Sony televisions.

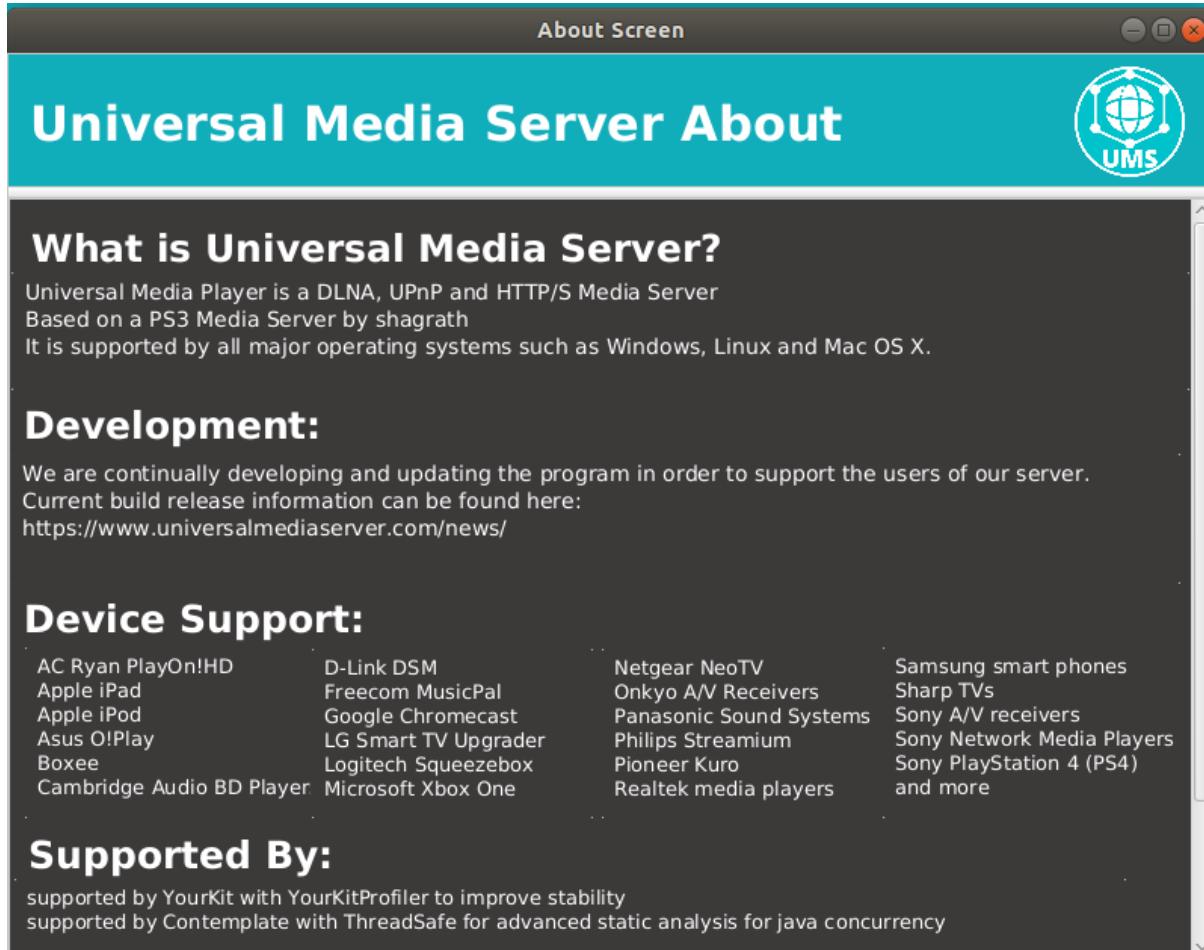
### Alternatives

Windows Media Player works well out of the box but isn't able to transcode or remux and has poor performance with a huge media database. Twonky media is not for free. Twonky media has its advantages in supporting a lot of different streaming clients, even NAS systems. TVersity is an average media server and has some advantages in streaming Internet content. But this advanced feature is not included in the free version. Mediatomb is often used on open source operating and embedded systems. It can be customized very well by experienced shellscript gurus. It is not a good choice for beginners.

< Previous Section      Next Section >

This is the high-fidelity design of the help screen. It has similar upgrades from the low-fidelity design as other screens with added colour and icons. The question mark icon is a symbol commonly used to represent help screens which is why I have used it here. Allowing the user to easily identify this screen as the help screen. I have also added functionality where the tab on the left is highlighted aqua. This hence allows the user to identify the help tab they currently have selected. Also, when the user selects the previous or next buttons at the bottom the highlighted tab on the left will also change to the corresponding previous or next help tab further helping the user in identifying the tab they are currently on.

About Screen:



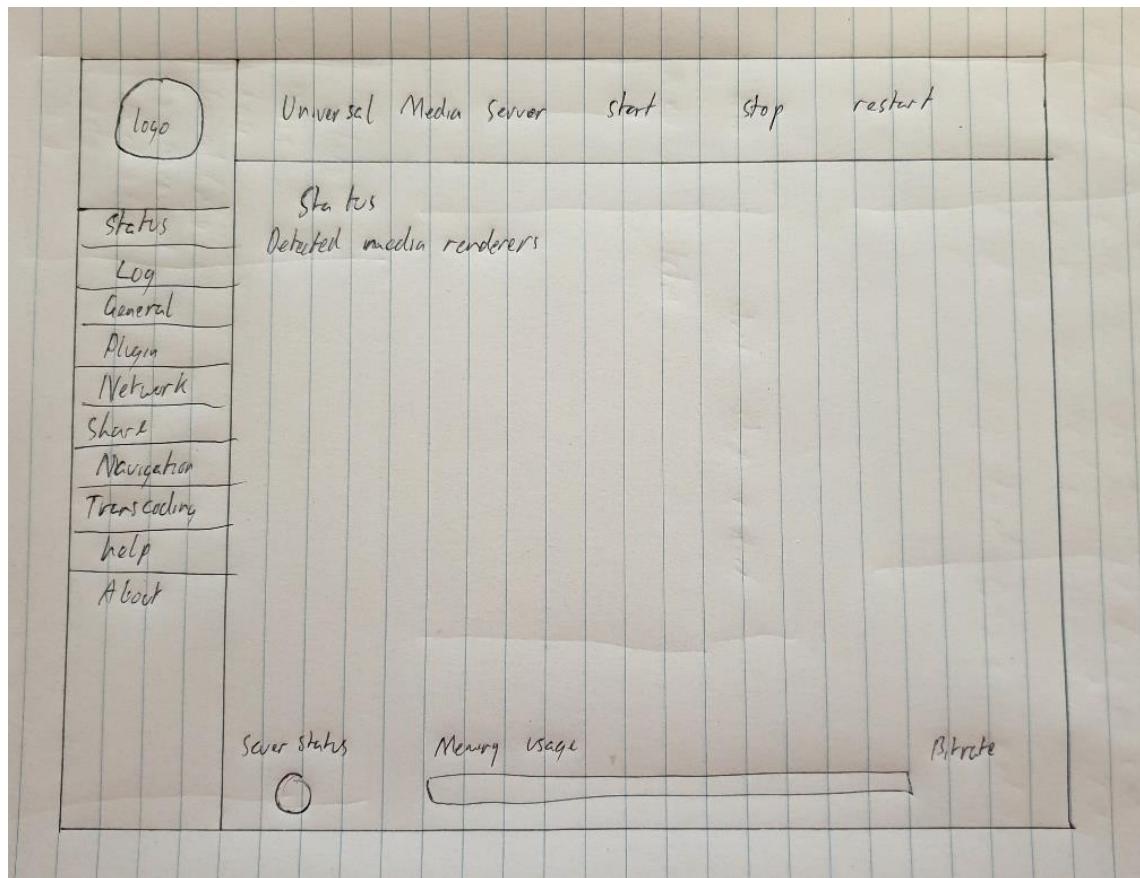
This is the high-fidelity design of the about screen. It is a very simple screen hence the changes from the original low-fidelity screen are very minimal. The main difference is the additional colour added giving what was previously a minimalist screen a more attractive look.

## 6.4 Brendan Scarfone

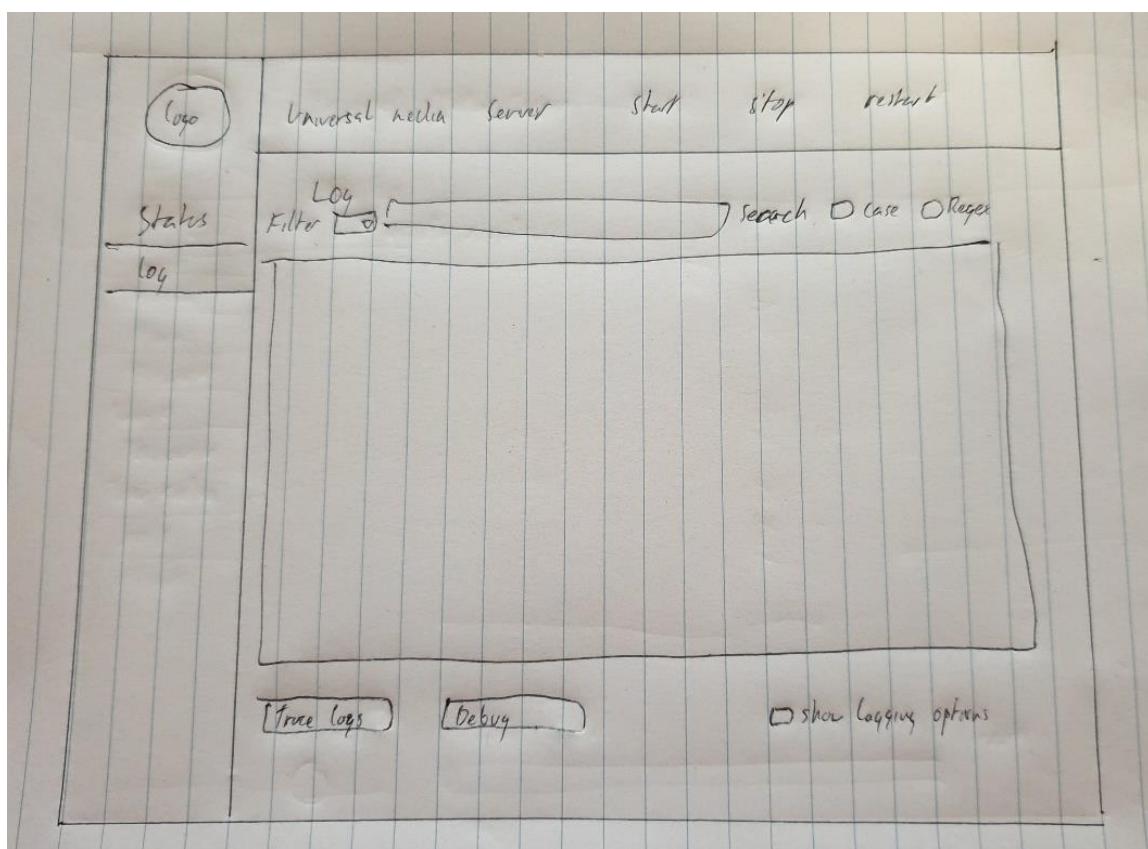
Contributor(s): Brendan

### 6.4.1 Low-fidelity Prototype

Status tab:



Log tab:



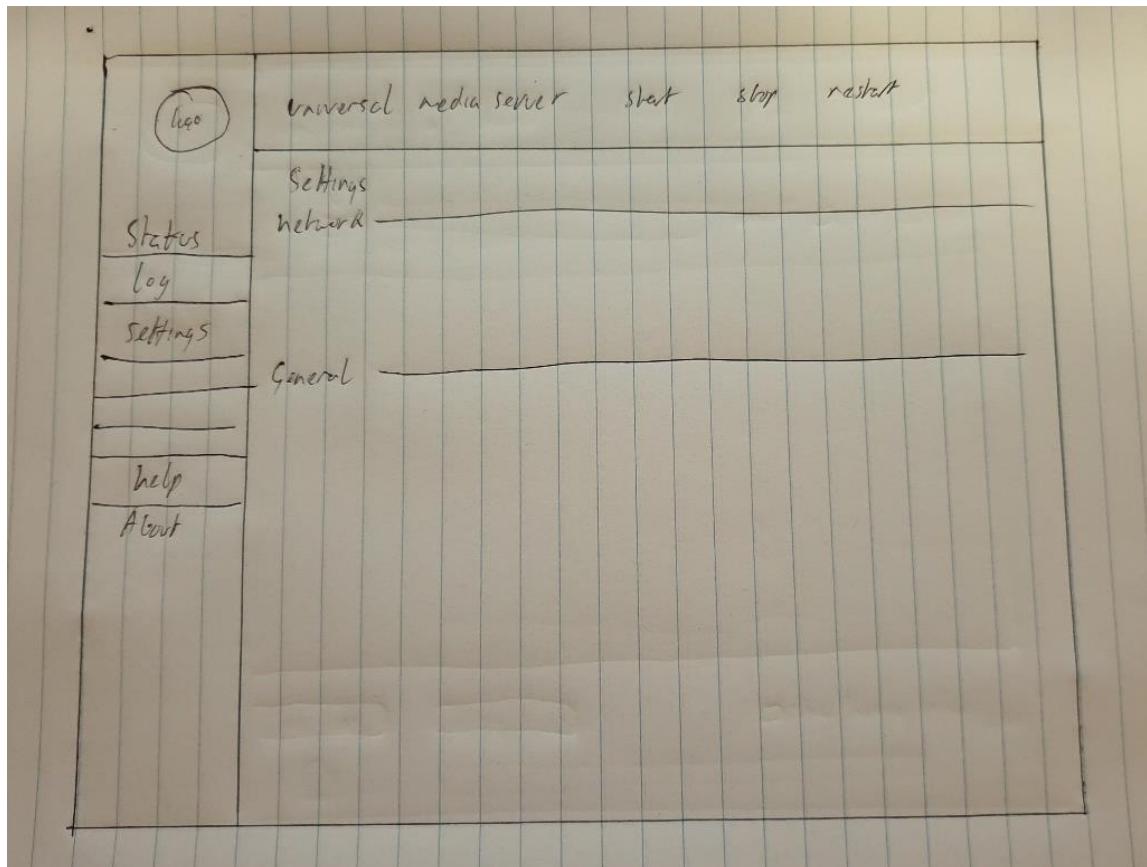
Help tab:

 <a href="#">Status</a> <a href="#">Log</a> <a href="#">Settings</a> <hr/> <a href="#">Help</a> <a href="#">Intro</a> <a href="#">What does it do?</a> <hr/> <a href="#">About</a> <a href="#">Alternatives</a>	Universal media server	start	stop	restart
	<a href="#">Intro</a>			
	<a href="#">What does it do?</a>			
	<a href="#">Help</a>			
	<a href="#">Alternatives</a>			

About tab:

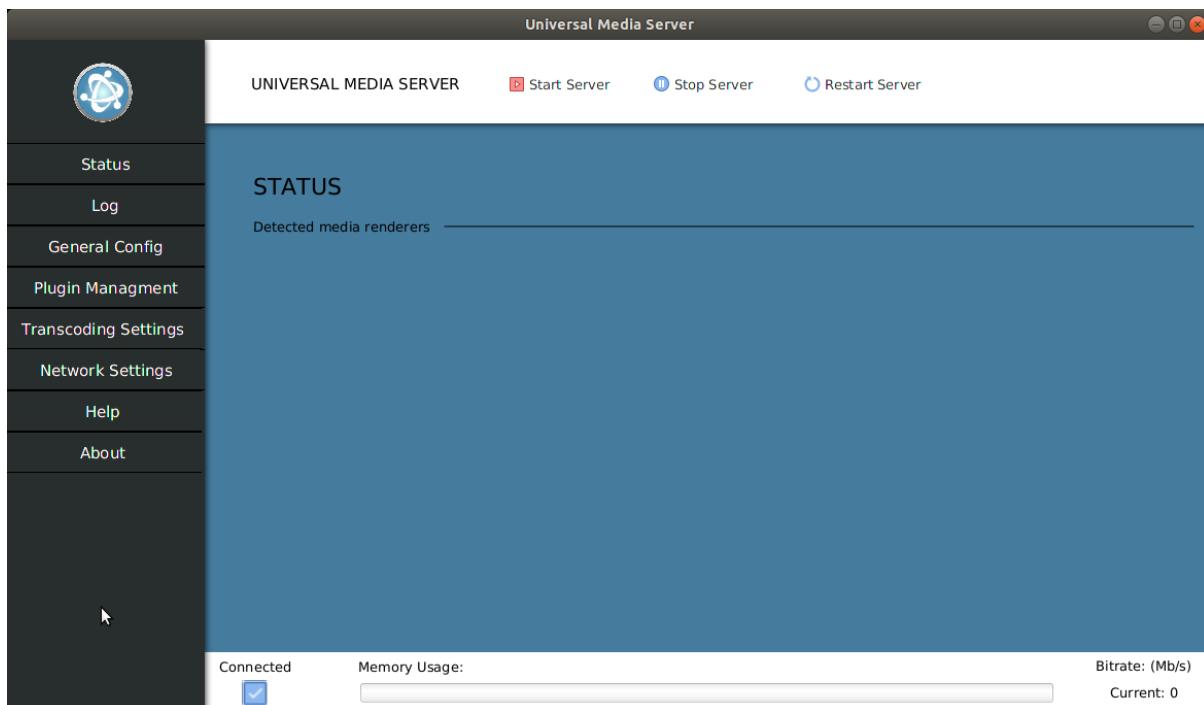
 <a href="#">Status</a> <a href="#">Log</a> <a href="#">Settings</a> <hr/> <a href="#">Help</a>  <a href="#">About</a> <hr/> <a href="#">Related links:</a>	Universal media server	start	stop	restart
	<a href="#">Universal media server</a>			
	<a href="#">Build date</a>			
	<a href="#">Link</a>			

Settings tab:



### 6.4.2 High-fidelity Prototype

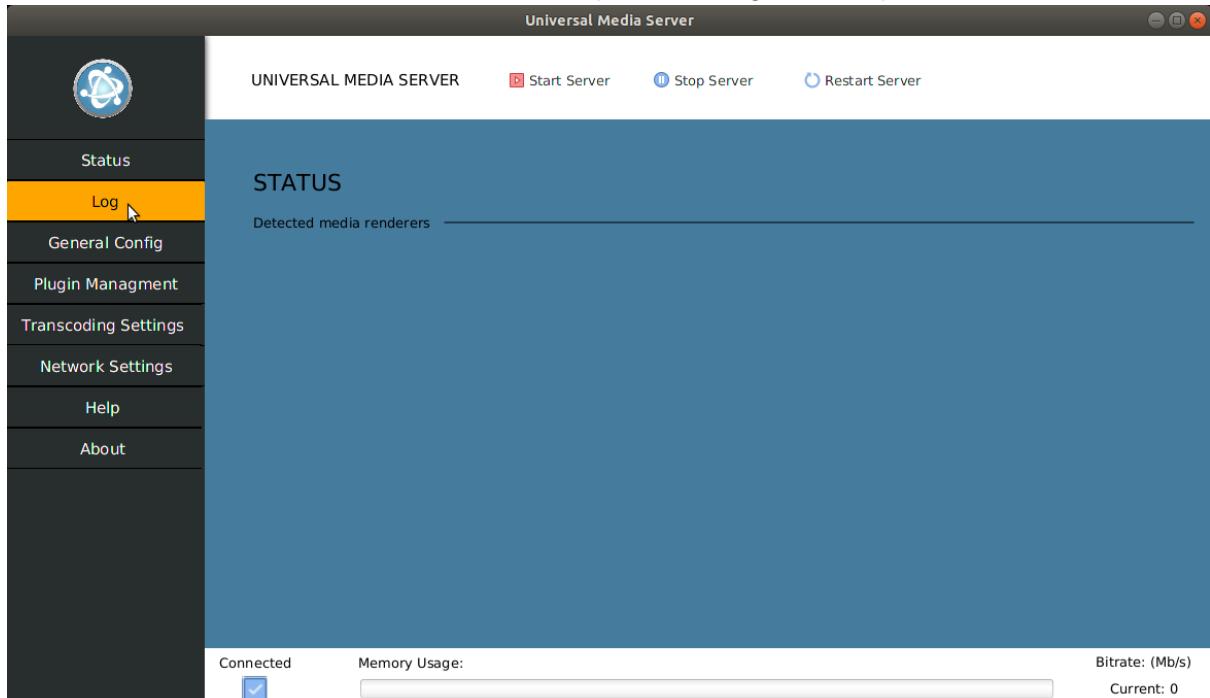
Status tab:



The program opens to the status page which shows the connected devices and the memory usage by the media server. The left side of the page allows the user to move between tabs which contain

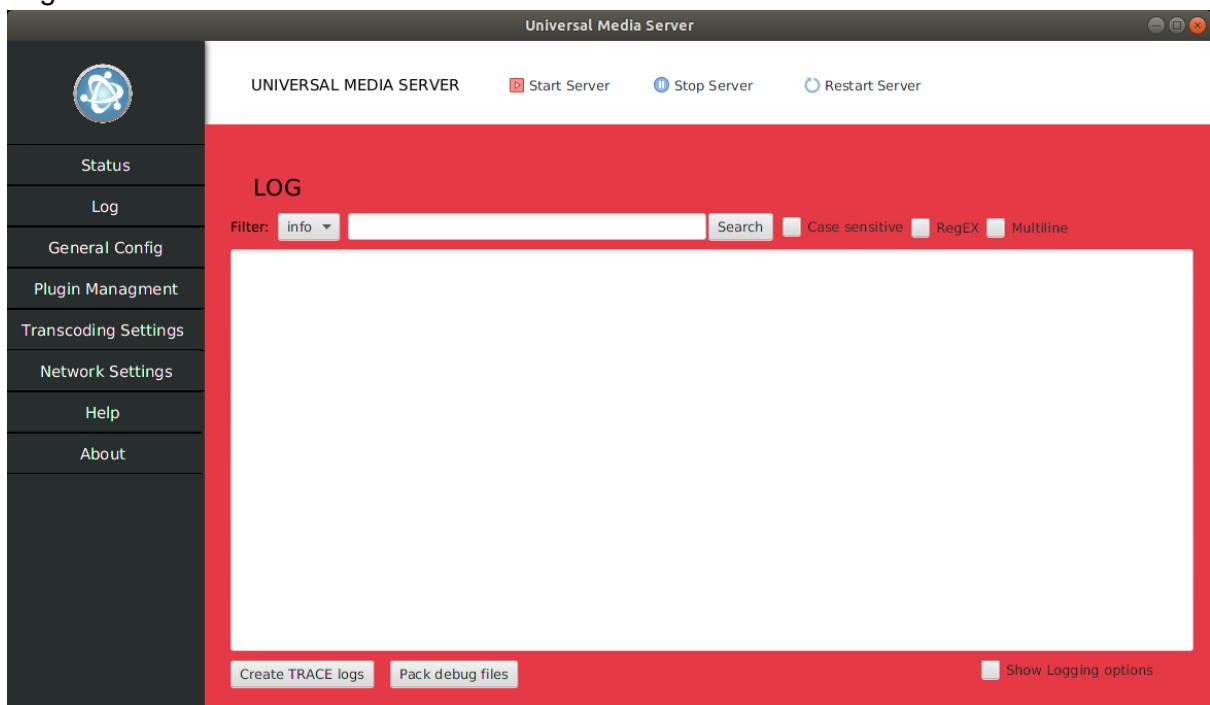
different options to customize the media server. The start server button at the top allows the user to start the server if it is not currently running which will be indicated by the server

status at the bottom left. The stop server button will stop the service but keep the program open. The server status will also change to disconnected. The restart server button will stop and start the server in case of an error or to update settings that require a restart.



Hovering the mouse over any of the tabs will highlight it to give some user feedback. It can then be selected to change to that tab.

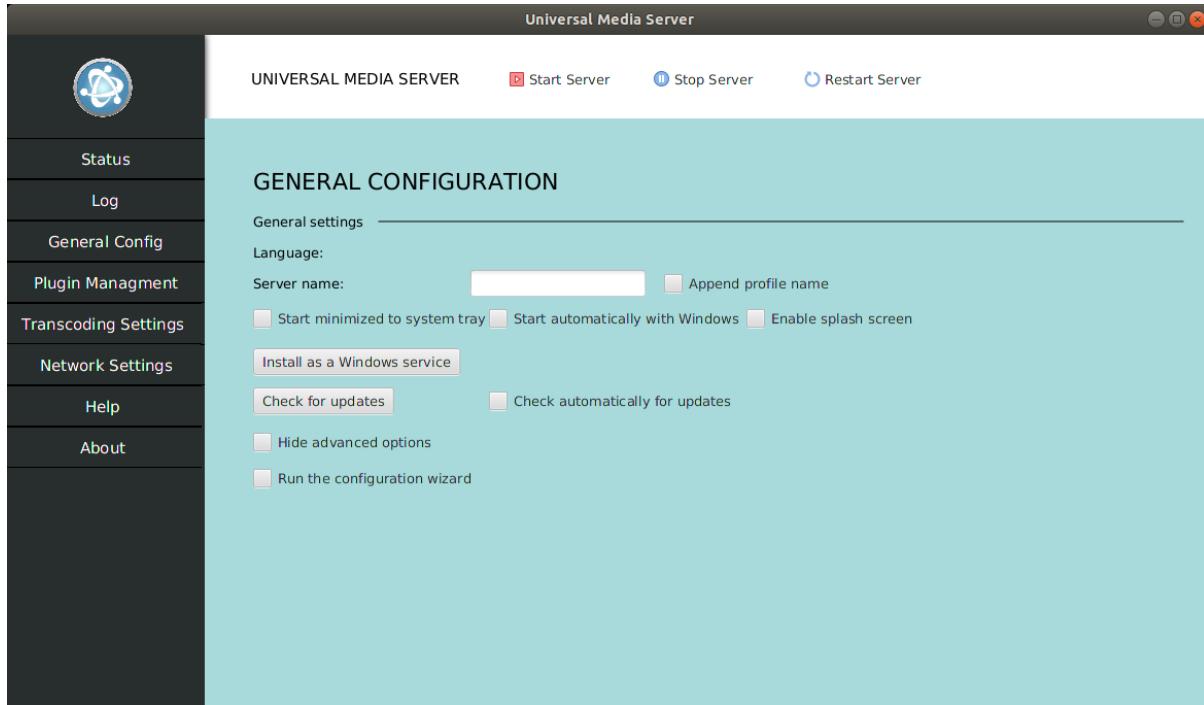
Log tab:



The log tab allows the user to view the log created by the media server inside the application. As well as having options to search the filter and search the log. The search function highlights the

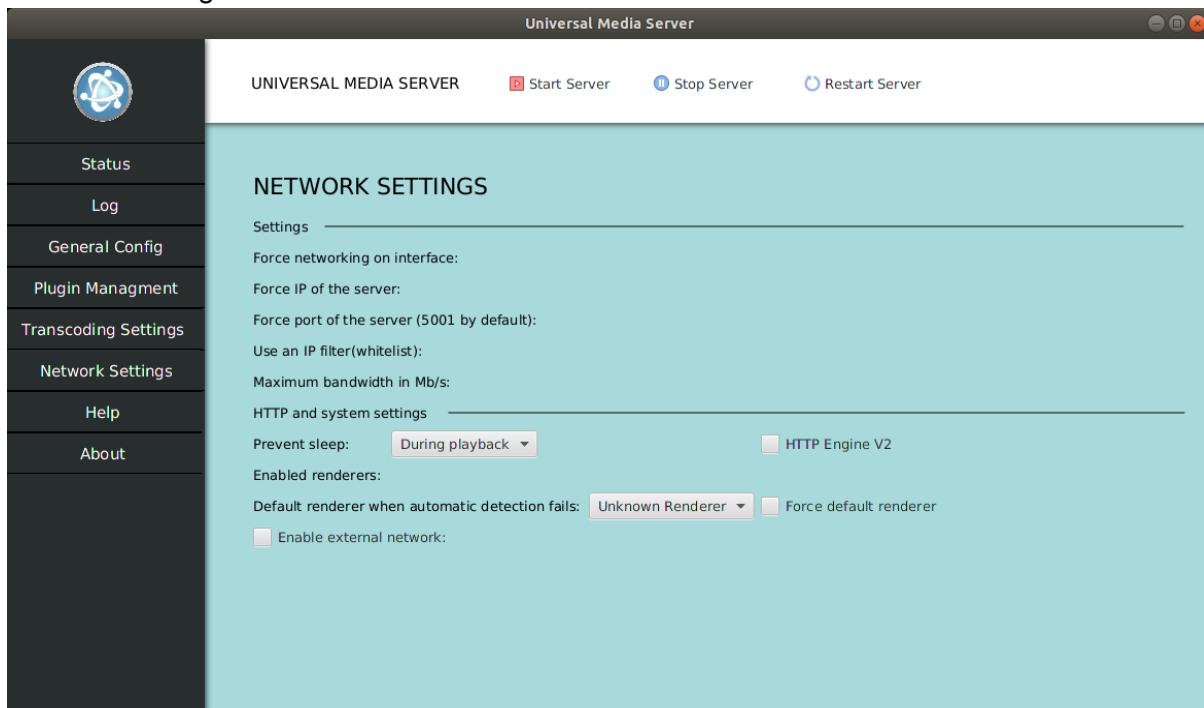
keywords and is affected by the check boxes on the right. The trace logs button allows the users to identify problems and easily report them to developers.

### General Config tab:



The general configuration tab includes the general settings such as changing the language of the application as well as choosing a server name. Each setting tab is separated into its relevant location to stop the confusion of many unrelated settings being in the same menu as it was in the original application. These will be separating into General settings, Network settings, Transcoding settings, Navigation and share. Each focused on one section of the operation of the media server.

### Network Settings tab:



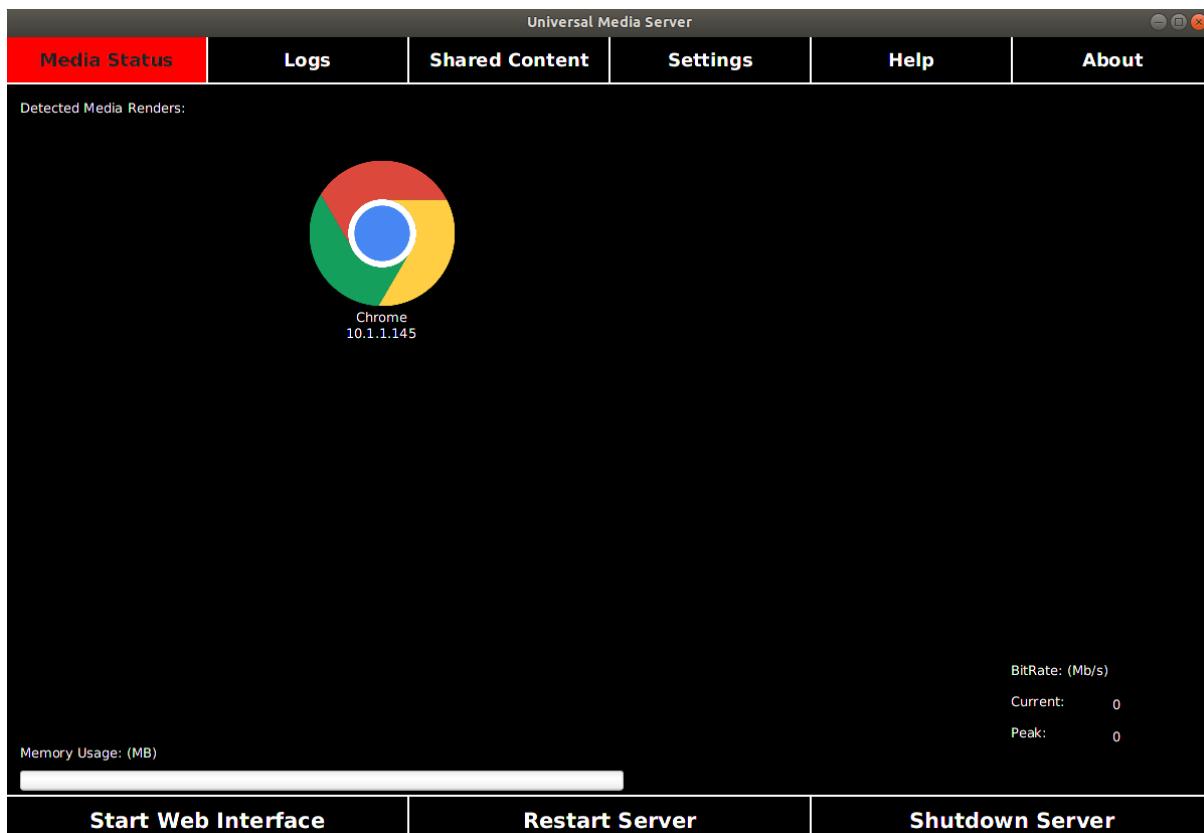
# 7.0 Implementations

## 7.1 Aidan Leavy

Contributor(s): Aidan

### 7.1.1 Implementation Discussion

The goal of this implementation was to create a more user friendly interface while still maintaining the wide array of functionality and user settings that the original program had. This implementation maintains the based navigation of the original but reduces the number of tabs visible by putting most of them in a secondary tab menu that can only be accessed by clicking on one of the primary tabs. This reduces the amount of clutter and helps to organise the page more than the original interface did.



Media Status screen

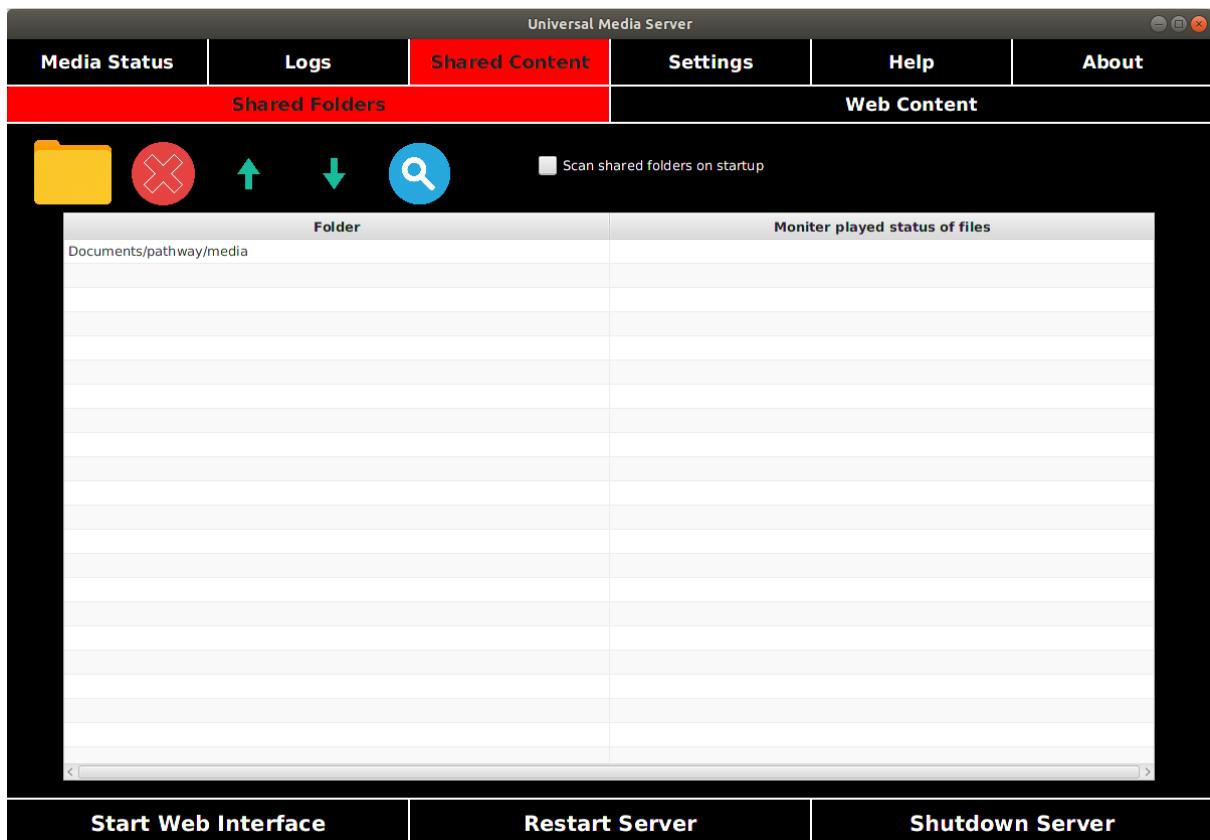
The only change made specifically to the Media Status screen is the addition of test data to simulate a media renderer being available to the server. This will create a button showing an icon generated by the program as it recognises the type of device along with the device's IP address if applicable and the device name.

The program wide change is the addition of white vertical bars placed between the buttons on the primary and final menu bars to complement the horizontal bars that are meant to further distinguish the buttons from each other and the background.

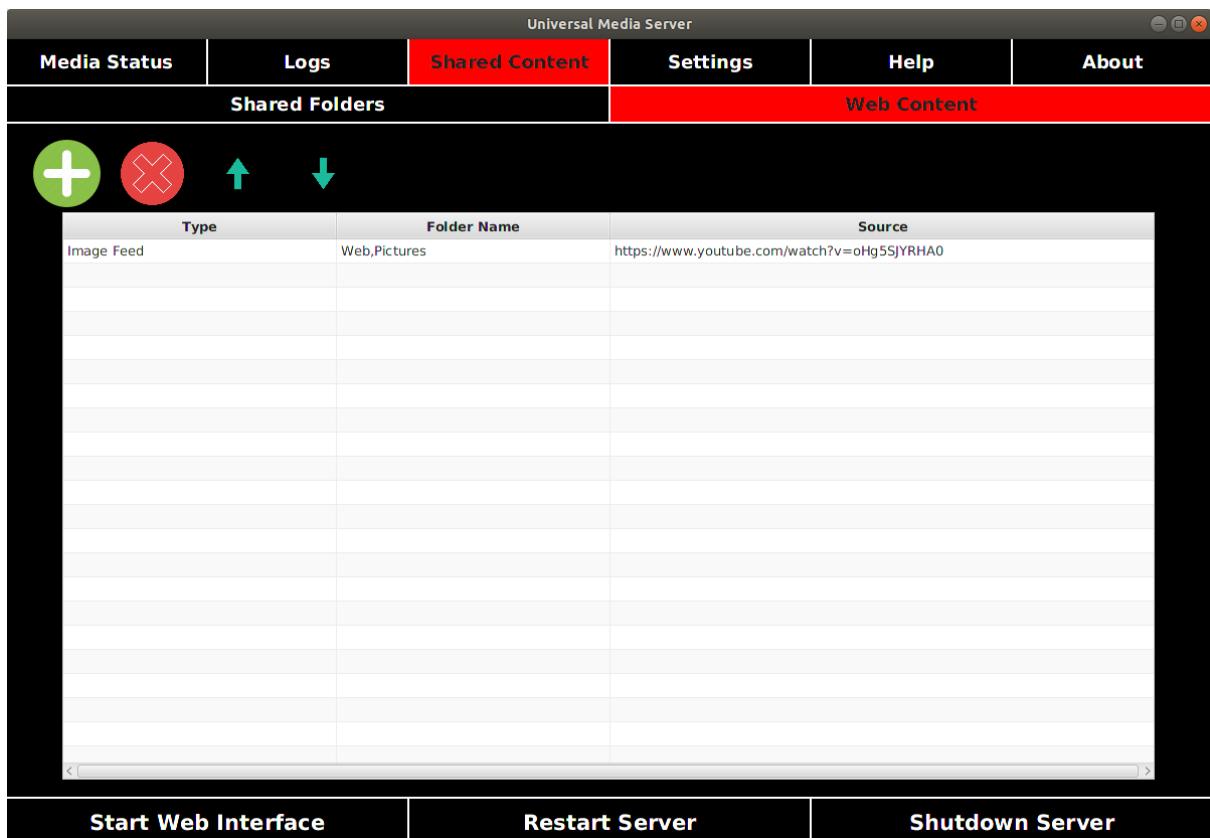


Logs screen

The Log Screen now has an improved menu for listing all the logs. The menu has a distinct title and the logs are displayed in rows with alternating backgrounds in order to distinguish them from each other. The dropdown menus now have default values stored in them as opposed to the high fidelity prototype which left them blank. The image shown has one log in the menu with two of the searching checkbox options selected and a dropdown menu open to display what they look like when in use.

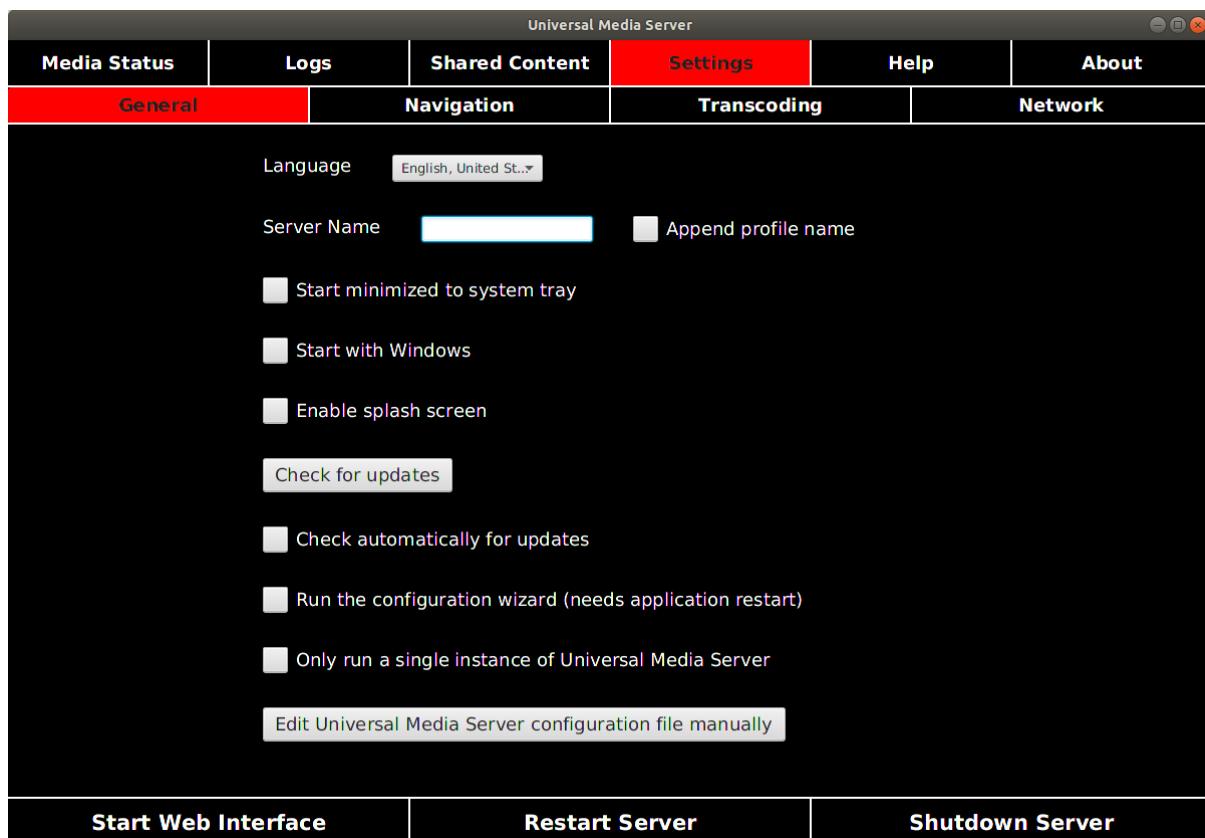


Shared Content screen - Shared Folders



Shared Content screen - Web Content

The only change made between the low fidelity prototype and the final implementation of the Shared Content screen was to improve the menu in which the folders are being listed by giving the table column titles and giving the rows slightly different alternating background colours to help identify the rows. The Web Content screen received a similar table and has the exact same functionality as described in the Shared Folders low fidelity prototype.



Settings screen - General

Universal Media Server

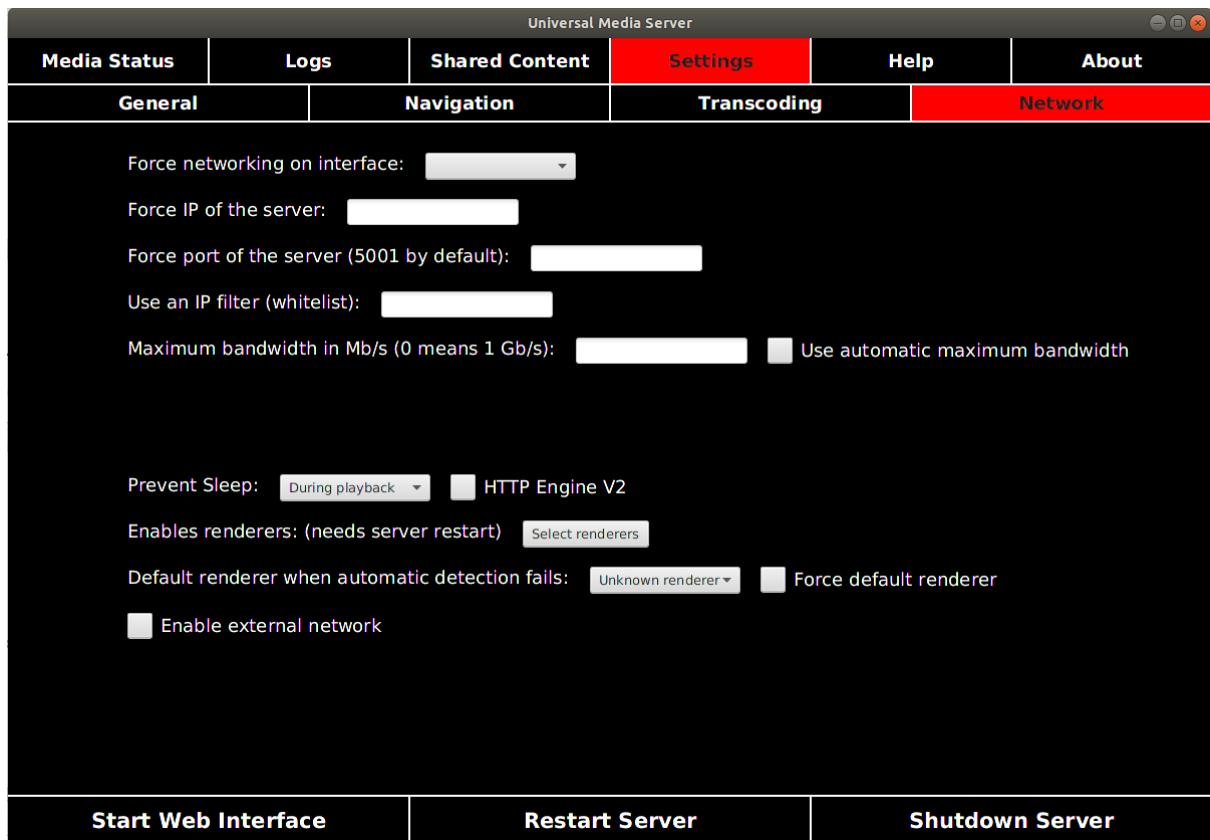
Media Status	Logs	Shared Content	Settings	Help	About
General	Navigation		Transcoding		Network
<b>Thumbnails</b>					
<input type="checkbox"/> Generate thumbnails	Thumbnail seeking position (in seconds)	<input type="text"/>			
Audio thumbnails import		Download from C...	<input type="checkbox"/> Use MPlayer for vvideo thumbnails	<input type="checkbox"/> Image thumbnails	
Alternate video cover art folder		<input type="text"/>	<input type="button"/>	<input type="checkbox"/> DVD ISO thumbnails	
<b>File sorting / naming</b>					
File order:	<input type="button"/> Alphanumeric	<input type="checkbox"/> Ignore articles "a" and "the" while sorting			
<input type="checkbox"/> Prettify filenames	<input type="checkbox"/> Hide file extensions	<input type="checkbox"/> Use info from www.OpenSubtitles.org			
Add subtitles information to video names:		<input type="button"/> None	<input type="checkbox"/> Add engine names after filenames		
<b>Virtual folders / files</b>					
<input type="checkbox"/> Show iTunes library	<input type="checkbox"/> Show iPhoto library	<input type="checkbox"/> Show Aperture library			
<input type="checkbox"/> Enable the Media Library	<input type="button"/> Empty the Media Library	<input type="checkbox"/> Show the "Media Library" folder			
<input type="checkbox"/> Browse compressed archives	<input type="checkbox"/> Show the "Server Settings" folder	<input type="checkbox"/> Show the "#--TRANSCODE--#" folder			
<input type="checkbox"/> Show the "Live Subtitles" folder	Minimum item limit before using A-Z folders	<input type="button"/>	<input type="checkbox"/> Show the "New Media" folder		
<input type="checkbox"/> Enable video resuming	<input type="checkbox"/> Show the "Recently Played" folder	<input type="checkbox"/> Hide Empty folders			
Fully played action	<input type="button"/>	<input type="button"/>	<input type="button"/>		
<b>Start Web Interface</b>		<b>Restart Server</b>		<b>Shutdown Server</b>	

Settings screen - Navigation

Universal Media Server

Media Status	Logs	Shared Content	Settings	Help	About
General	Navigation		Transcoding		Network
<b>General Transcoding Settings</b>					
Maximum transcode buffer size in MB: (Recommended 200) <input type="text"/>					
CPU threads to use when enabled for engine: <input type="button"/> 4					
<input type="checkbox"/> Chapters support in the "#--TRANSCODE--#" folder (minutes) <input type="text"/>					
<input type="checkbox"/> Disable subtitles					
<b>Video Settings</b>					
<input type="checkbox"/> Enable GPU acceleration					
<input type="checkbox"/> Lossless DVD video playback					
Transcoding quality (MPEG-2) <input type="button"/> Automatic (wirele...)					
Transcoding quality (H.264) <input type="button"/> Automatic (wirele...)					
Skip transcoding for the following extensions (comma separated) <input type="text"/>					
Force transcoding for the following extensions (comma separated) <input type="text"/>					
<b>Audio Settings</b>					
Maximum number of audio channels to output for Ac-3 re-encoding: <input type="button"/> 6 channels (5.1)					
<b>Start Web Interface</b>		<b>Restart Server</b>		<b>Shutdown Server</b>	

Settings screen - Transcoding



Settings screen - Network

None of the Settings screens were changed beyond the initial prototyping concept implemented only in the General page. The secondary menu bar received the same vertical bar treatment that the primary and final menu bars did to help make them visually distinct from each other.

Universal Media Server

Media Status		Logs		Shared Content	Settings		Help	About	
Home	Introduction	Networking and technical basics		Installation and basics configuration	General configuration	Navigation/Share settings	Transcoding settings		
VLC media player	AviSynth	CoreAVC		Command line arguments	Plugins	Frequently Asked Questions (FAQ)	Useful links		

**What does it do?**

Universal Media Server (or UMS for short) is a program that streams media like video, audio or images from your computer or the web to a device called a "renderer" that can be anything that supports DLNA like a game console, a TV, a computer software or a mobile phone.

This might sound simple, but it is in fact a complex matter. There are dozens of media formats in the world, and your console or TV only understands a couple of them. UMS allows you to browse media on your favorite renderer without having to worry about the details of what you are viewing. UMS takes care of retrieving the media (from file, network or web) and translating it to the best format that your device understands.

**Media renderers**

UMS is a DLNA compliant UPnP Media Server. It was originally written to support the PlayStation 3. However, other renderers support the same standards as the PlayStation 3 and UMS has been expanded to support a range of them, including Xbox 360 and various Samsung and Sony televisions.

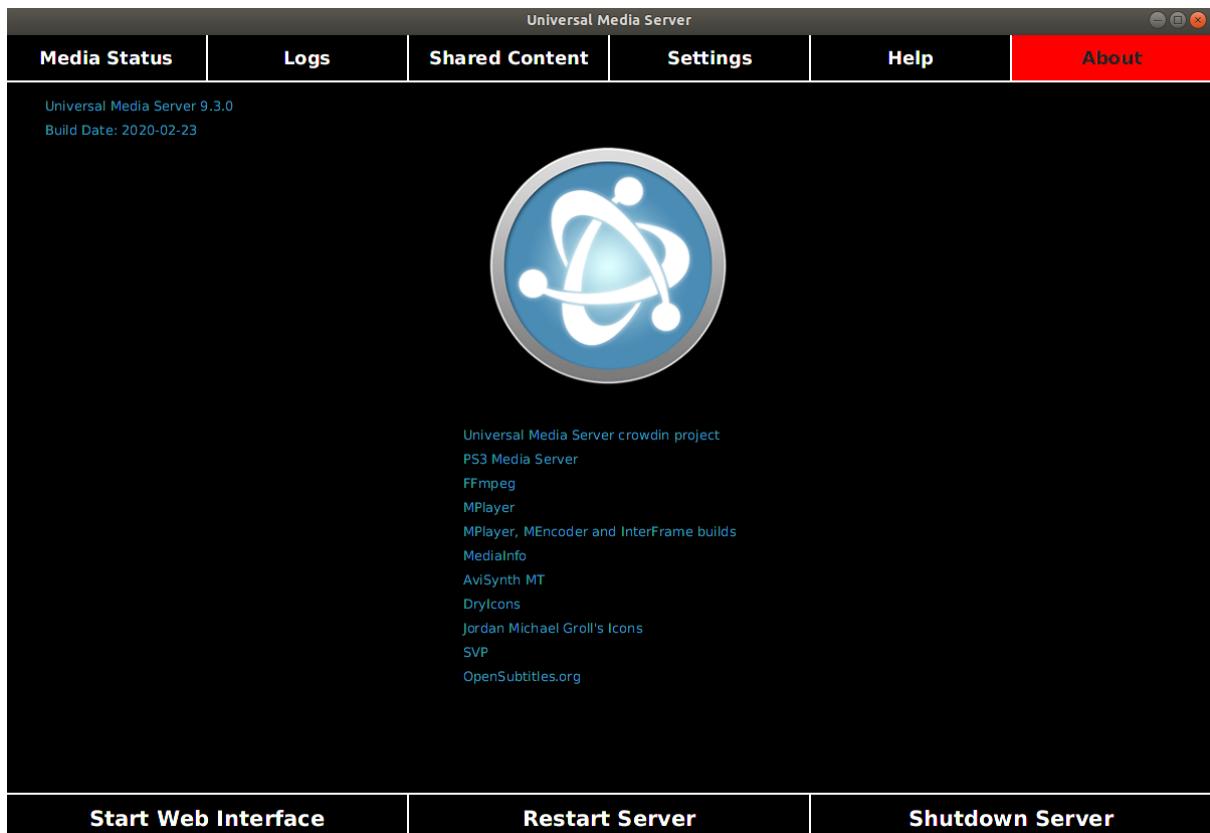
**Why UMS?**

UMS is available for Windows, Linux and Mac OS X. A prerequisite is the installation of Java Runtime Environment (JRE) because UMS is programmed in Java. UMS is shipped with all required software to make it work out of the box. However, additional features may require the installation of additional software. There are a lot of technical details involved in the process of translating media. UMS tries to minimize this for the user by shipping with working defaults for most situations. You simply start UMS and it will find your device and start communicating with it without any extra configuration. UMS is optimized for use with the Playstation 3. But you can also use an Xbox 360, Popcorn Hour and various Samsung or Sony TVs as streaming clients. In the old days, people would use a conversion program like "mkv2vob" and connect USB hard drives with converted files to the PS3. With UMS there is no need for converting files any more. If converting is necessary, UMS will do it for you on the fly. UMS also makes it possible to stream web content like Youtube, GameTrailers, web radio or Picasa to your device. UMS is free Open Source software and has been designed to allow third party plugins to extend its functionality. You are welcome to help improve UMS or write a plugin for others to use. See the project homepage for more information. The most important features of UMS are transcoding and muxing.

**Start Web Interface**      **Restart Server**      **Shutdown Server**

### Help screen - Introduction

The Help screen received the largest overhaul out of all the original prototypes. As seen above, the scroll menu meant to navigate the different help categories is gone and has been replaced with a secondary menu bar that has two layers on it, containing a total of 14 buttons. This was done so that the Help screen would be more consistent with the other screens in that any need to change the screen while remaining on the help screen was done using a secondary menu bar since that is the method implemented every other time the problem has arisen. By maintaining that consistency, it is hoped that this makes it more intuitive for users as they would already be used to the tabbing method of navigation by using the other tabs previously.



About screen

The about screen made no changes from its low fidelity prototype, though the prototype didn't actually display any text or logos and simply relied on placeholders meant to symbolise them instead for simplicity. The blue text consists of links that will take the user to their respective web pages when clicked to give the user more information about each topic.

### 7.1.2 User Involvement

When the user starts the application, they will be taken to the Media Status tab which allows them to view any connected media renderers. In the current implementation, the only media renderer available is Chrome which is implemented using to show the user what it looks like to have a renderer connected. In the final implementation, this screen will automatically update itself with all media renderers that become connected to the server. From this screen, the user can also see the current memory being used as well as the current and peak bitrates. These cannot be interacted with and are all set to 0 as there is no functioning server to measure data from. From here, the user can either select one of the tabs at the bottom, none of which are implemented or have any user feedback, or select one of the tabs on the top row which will turn red when pressed and take the user to the screen corresponding with that tab.

On the Logs screen has 4 checkboxes that can be interacted with but have no implementation. There are two functioning dropdown menus and a text field that acts as a search bar that the user can interact with, though they do not do anything. The user can click on and highlight the logs with the log table, though this has no effect and isn't supposed to in the final implementation.

The Shared Content screen has two subcategories that can be accessed by the user in the secondary menu bar. These two tabs act just like the primary bar in that they turn red when clicked and stay that way until a different tab is pressed. On both the Shared Folders and Web Content tabs, there are icons between the table and secondary menu bar that do nothing when clicked, though they do create a tooltip when hovered over describing their intended function. The folders/web content in their respective tables can be clicked and become highlighted. This is necessary for the final implementation of most of the buttons that need some content highlighted in order to determine the target of their functionality.

The Setting screens are all accessible through the functional secondary menu bar and the user can change most of the settings. Some of the settings have buttons with the text "...". These buttons are intended to open the operating systems file manager and allow the user to select a file/folder from the computer to be the target of one of the settings, though this is not implemented. The Transcoding tab has a scroll wheel to gain access to all of the settings as there are too many to cramp onto one page without it, which the user can interact with by either using the scroll wheel on their mouse or by clicking on the scroll menu and dragging it.

The Help screen has 14 tabs in its secondary menu bar, though only two of them have any information in them. These are the first two tabs, Introduction and Home. These two pages have text that can be read, but none of it will be intractable even in the final version, except for a scroll menu if that is needed to access the information on a page. Despite all the other tabs being empty, they all turn red when clicked to signify that they are all accessible by the user.

The hyperlinks on the About screen receive a highlighted border to signify that they have been clicked but do not open the links they are intended to. There is nothing else on this screen that is intended to be interacted with and the Universal media Server logo is just an icon that is present for aesthetic reasons.

Due to this being a limited GUI that doesn't have any functionality attached to it, most of the buttons do not have any functionality to them. The buttons will glow blue to signify that they have been clicked and provide some user feedback but nothing else will happen. The checkboxes will get ticked when clicked on and dropdown menus will provide the full list of options and display the value chosen once finished. There are several text fields where users can type information into, and they will respond to any text typed into them and maintain it once the text field is exited. All settings are deleted if a user swaps to a different tab in either the primary or secondary menu bars.

### 7.1.3 Test Cases

1. Detect media devices
  - a. Run the program
2. Changing screens
  - a. Run the program

The program will automatically find media renderers and list them on the Media Status screen. This will not occur in the current implementation as it is running dummy code.

- b. From any screen, press any of the tabs on the menu bar at the top of the screen to change tabs

All menu tabs will open the correct screen when clicked.

3. Quit the application

- a. From any screen, press the exit button at the top right

The exit button will exit the program

4. The user wants to find subtitles for a file

- a. Press the About tab
- b. Locate and press the OpenSubtitles.org link

Since the hyperlinks are non-functional, no web page will be opened

5. The user wants to add a file to the shared files

- a. Press the Shared Content tab in the primary bar
- b. Press the Open New Folder button beneath the secondary menu
- c. Select folder from the operating systems file manager popup

Since the Open New Folder button is not implemented, nothing will happen when it is pressed

6. The user wants to remove a file from the web content

- a. Press the Shared Content tab
- b. Press the secondary Web Content tab
- c. Highlight the web content in the table to be removed
- d. Press the Remove Web Content button

The Remove Web Content button is not implemented so this fails, but every other step will work.

7. The user wants to search the log

- a. Press the Log tab
- b. Enters the String they want to search in the search bar
- c. Select the appropriate options to narrow down the search
- d. Presses the search button

The search button is not implemented; thus, it will not produce any results. This also hides the fact that none of the optional search parameters are not implemented and won't do anything either.

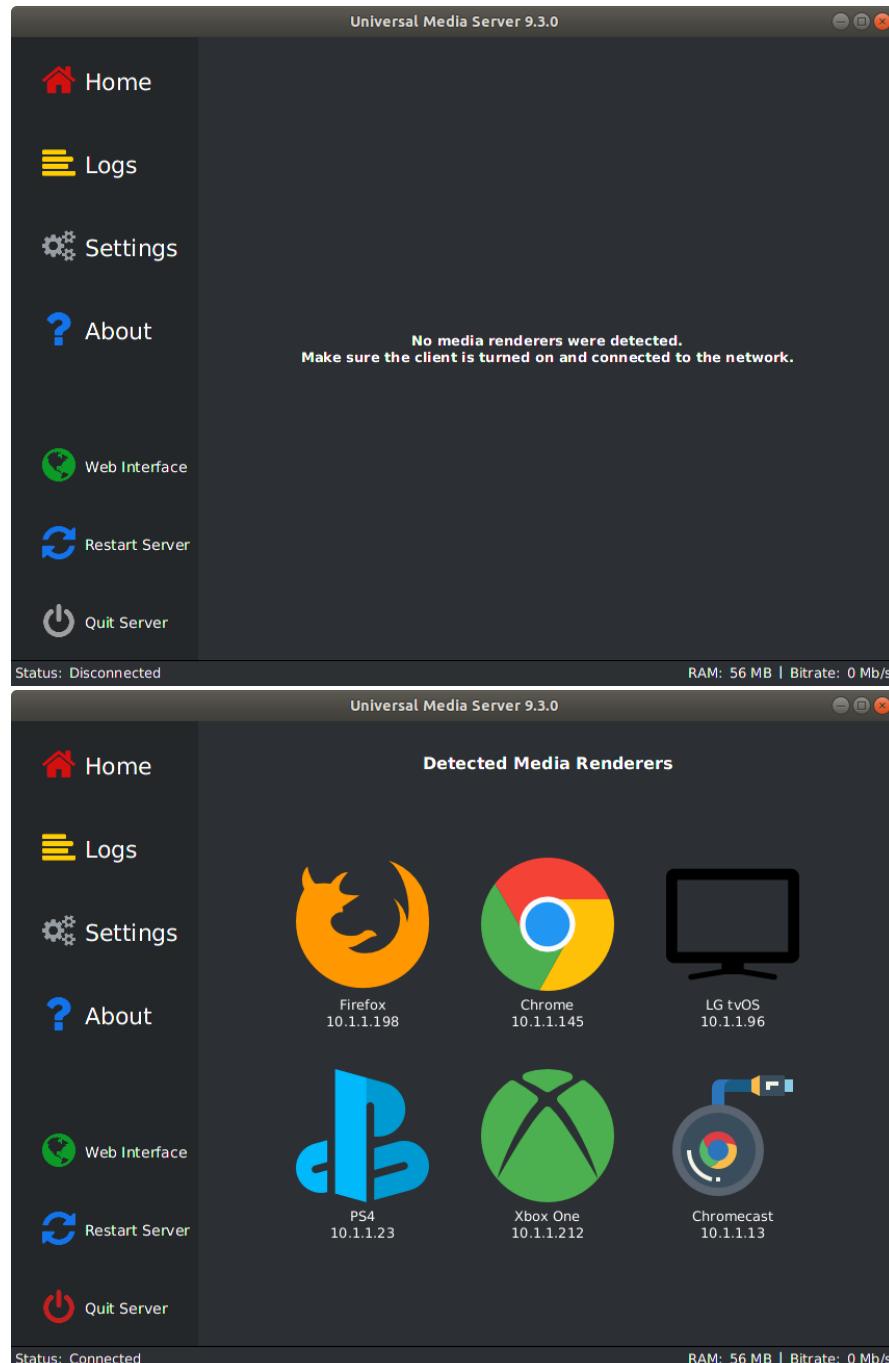
## 7.2 Lachlan Mackenzie

Contributor(s): Lachlan

### 7.2.1 Implementation Discussion

For my implementation, my main goal was to create a clutter free, modern user interface. The main idea behind this was the choice to instead of having a fully tabbed layout like in the original program, have a simple sidebar on the left which contains all of the navigation the user could need. This type of design was inspired by many programs that I frequently use such as Plex, Discord, and Spotify.

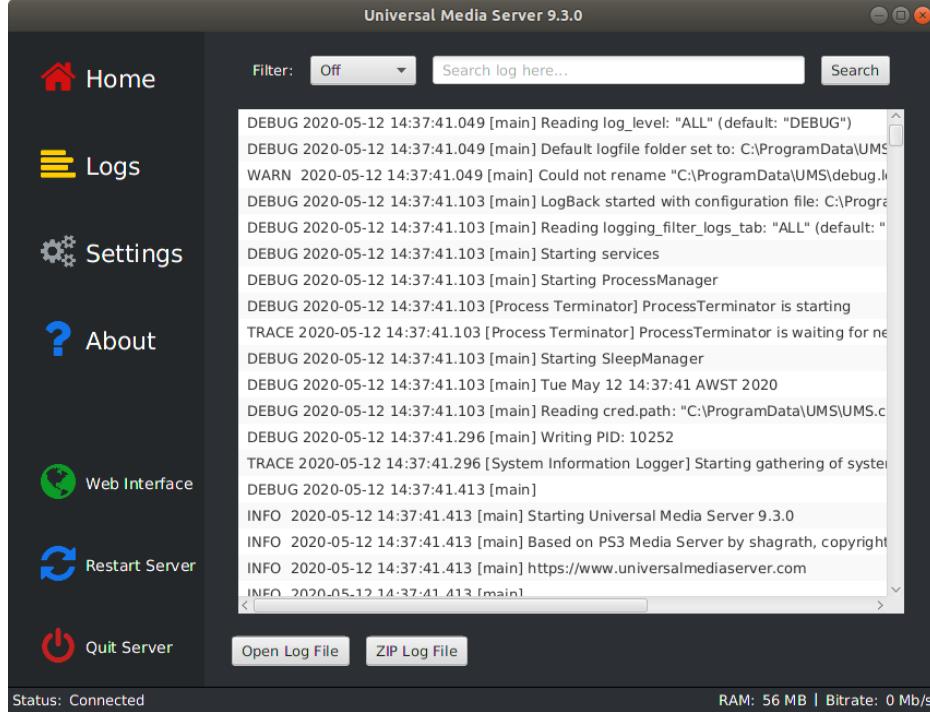
#### Home Screen



The main change for the home screen is the icons used. Originally in the high-fidelity prototype I used SVG images as I came across them first. However, in the implementation

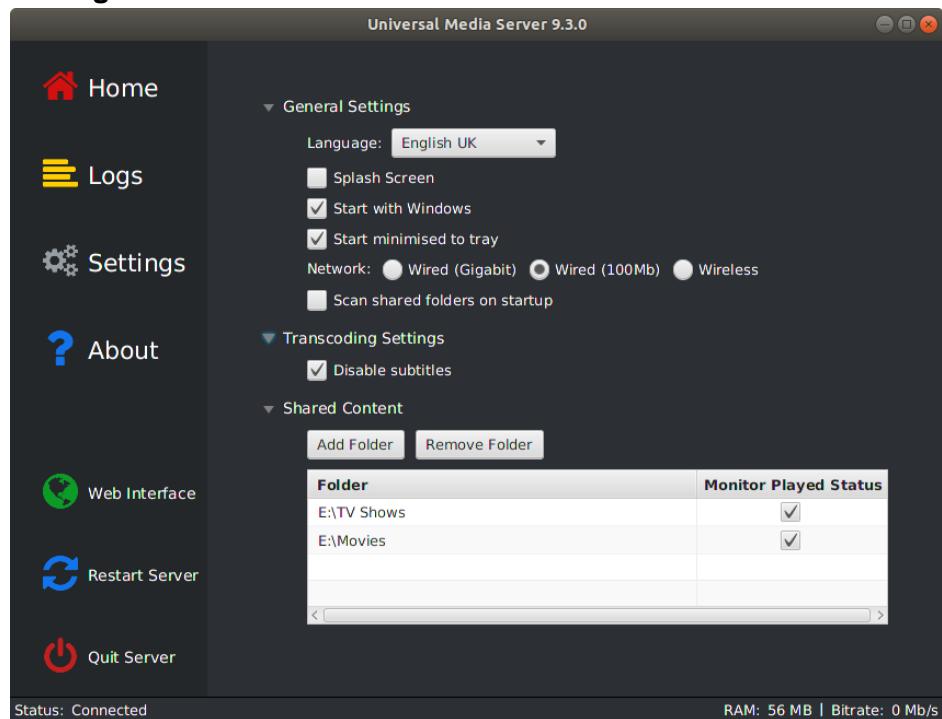
using SVGs for images in JavaFX cannot be easily done. Therefore, I ended up using PNG icons from [www.flaticon.com](http://www.flaticon.com). The other difference is the removal of the separator in the sidebar separating the navigation buttons from the server buttons. I decided to remove this as the empty space that is already there does a good job at providing the visual separation needed for the user. Regarding the navigation buttons, I felt it was a better decision to swap the position of the Settings and Logs buttons so that Logs is above Settings.

## Logs Screen



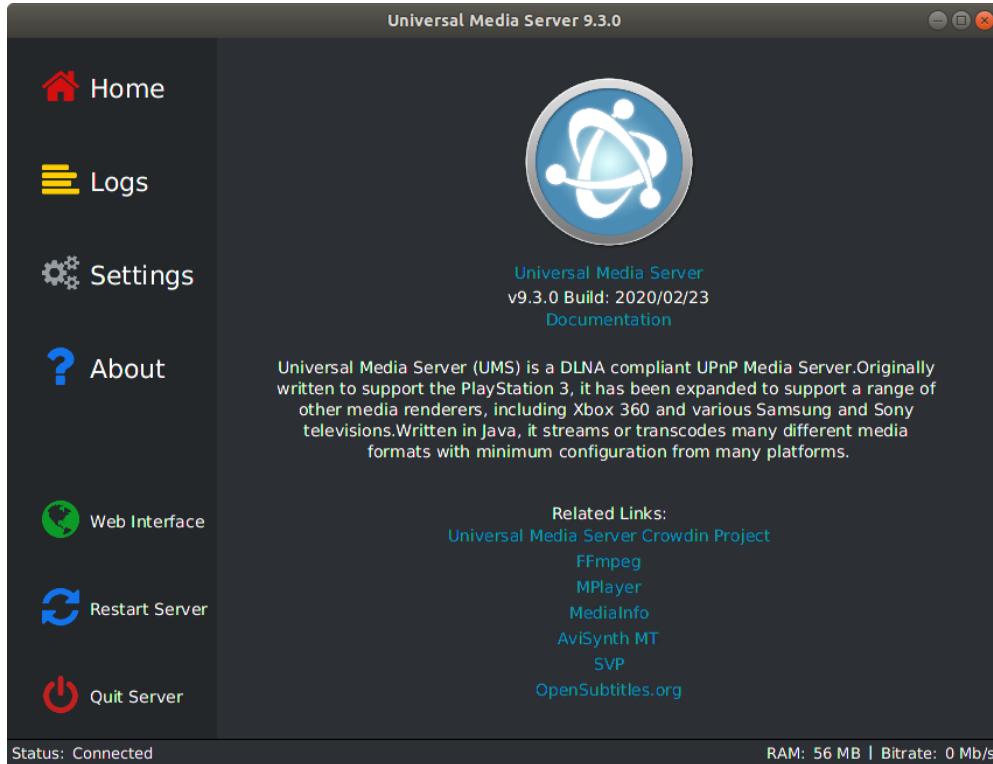
For the logs screen the biggest change from the prototypes is the styling of the log file viewer. Originally it also incorporated the dark theme that is throughout the design, but I found it was unfeasible to achieve the correct styling that I had in my vision. Hence, the default ListView styling is being used instead. The top section has also been changed slightly, the log level filter ComboBox has been moved to the left-most of the three controls instead of the middle. This decision made sense for me as it mimics how the actual log file is formatted with the log level being the first part of each line.

## Settings Screen



In the original high-fidelity prototype, I had included buttons inside the Shared Content table for each entry. To achieve the correct functionality with these buttons it would be a massive time sink and unfeasible for me to implement. As an alternative I decided to develop two new buttons residing just above the table. The Add Folder button will open the operating system's directory chooser for the user to select a directory while the Remove Folder button will remove the entry that the user has currently selected in the table.

## About Screen



For the about screen, I decided to add a simple blurb about Universal Media Server that I retrieved from their website. The related links section had some removed in my implementation as they no longer pertained to me as some of them were links to the icons used in the original GUI. All of those links will open the user's default browser for viewing. Additionally, instead of a separate Help section within the program I decided to instead link directly to the documentation website that was previously embedded in the original program. Using the browser for the documentation will make it much easier for the user to navigate the documentation as it is likely they will already be familiar with one.

### 7.2.2 User Involvement

Once the user starts the program the Home screen will be loaded. If there are no media renderers detected the user will be notified via a message on the Home screen explaining why the screen doesn't have any content on it and what they can do to possibly remedy any issues. Otherwise when media renderers are detected at start up the screen will be filled in a grid like fashion display the icon, name and IP address of the renderer.

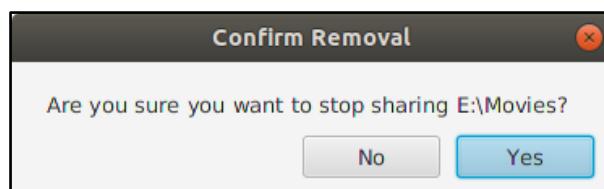
When the user hovers over these renderer icons with their mouse, the mouse style will change to indicate that it is clickable. If the user clicks any of the icons, media player controls should popup in a new window to allow the user to play their shared content. However, that part of the GUI has not been implemented. If at any time a new renderer is detected, the screen will be automatically updated to contain it.

When the user hovers over the different buttons on the sidebar their respective backgrounds will change to give feedback to the user that these are clickable. If the user is unsure about

what the server buttons at the bottom left might do, they can simply hover over them to pop up a tooltip to give extra information pertaining to that button. Additionally, when the user chooses to use the Quit Server button another confirmation dialog will appear, once confirmed by the user the quit icon will change from a red colour to a grey which indicates to the user that it was successful in shutting down. The Restart Server button also has a similar effect. When it is pressed, the icon will change from a blue to a grey colour, once the server has restarted the colour will return to its normal blue.

When the user selects the Logs button in the sidebar, the Logs screen is loaded. This screen will automatically load in the currently used log file and display it to the user ready to be searched. The user can choose to filter the search by log level by selecting a value in the ComboBox on the left. Once the user starts to type their search term into the search bar, the log file will automatically be filtered for the log level and search term on every key pressed. The user can also click the button in the bottom left to open the actual log file in the default viewer for that extension type or they could select the ZIP Log File button to compress the log file to be sent off to the developer for debugging but this is just stubbed with a simple dialog for now.

When the user selects the Settings button in the sidebar, the Settings screen is loaded. The Settings screen contains multiple collapsible sections which each contain separate settings depending on their category. For the first two sections the user can select values from a ComboBox, check boxes or even select an option from a radio group. The third section of the settings is the Shared Content section. This section allows the user to add or remove directories that they would like to be able to share across to any of the detected media renderers. Firstly, the Add Folder button will prompt the user to choose a folder containing the content to be shared. This is done by using the operating system's directory chooser dialog. Secondly, the user can remove a folder by clicking the entry in the table and then selecting the Remove Folder button. When the user clicks this button a confirmation dialog will appear, as shown below, for the user to confirm that they are sure of their decision.



All of the settings currently in this program are persistent, meaning that any changes made by the user will be saved to a file for later use. Therefore, the user won't have to re-enter their custom settings each time the program is launched.

When the user selects the About button in the sidebar, the About screen is loaded. From this screen the user can visit many different links by clicking on them. The links will change when hovered over and when they have been visited at least one. This page also contains the link to the documentation website that the user can easily visit by clicking one link. This link will be opened in the user's default browser.

The program also responds to the user resizing the window. Each screen was developed with that in mind.

Universal Media Server 9.3.0

**Detected Media Renderers**

- Firefox 10.1.1.198
- Chrome 10.1.1.145
- LG tvOS 10.1.1.96
- PS4 10.1.1.23
- Xbox One 10.1.1.212
- Chromecast 10.1.1.13

**Logs**

Universal Media Server 9.3.0

Filter: Off Search log here... Search

```

DEBUG 2020-05-12 14:37:41.049 [main] Reading log_level: "ALL" (default: "DEBUG")
DEBUG 2020-05-12 14:37:41.049 [main] Default logfile folder set to: C:\ProgramData\UMS\debug
WARN 2020-05-12 14:37:41.049 [main] Could not rename "C:\ProgramData\UMS\debug.log" to "C:\ProgramData\UMS\debug.log.prev"
DEBUG 2020-05-12 14:37:41.103 [main] Logback started with configuration file: C:\Program Files (x86)\Universal Media Server\logback.xml
DEBUG 2020-05-12 14:37:41.103 [main] Reading logging_filter_logs_tab: "ALL" (default: "INFO")
DEBUG 2020-05-12 14:37:41.103 [main] Starting services
DEBUG 2020-05-12 14:37:41.103 [main] Starting ProcessManager
DEBUG 2020-05-12 14:37:41.103 [Process Terminator] ProcessTerminator is starting
TRACE 2020-05-12 14:37:41.103 [Process Terminator] ProcessTerminator is waiting for new tickets
DEBUG 2020-05-12 14:37:41.103 [main] Starting SleepManager
DEBUG 2020-05-12 14:37:41.103 [main] May 12 14:37:41 AWST 2020
DEBUG 2020-05-12 14:37:41.103 [main] Reading cred.path: "C:\ProgramData\UMS\UMS.cred" (default: "")
DEBUG 2020-05-12 14:37:41.296 [main] Writing PID: 10252
TRACE 2020-05-12 14:37:41.296 [System Information Logger] Starting gathering of system information
DEBUG 2020-05-12 14:37:41.413 [main]
INFO 2020-05-12 14:37:41.413 [main] Starting Universal Media Server 9.3.0
INFO 2020-05-12 14:37:41.413 [main] Based on PS3 Media Server by shagrath; copyright 2008-2014
INFO 2020-05-12 14:37:41.413 [main] https://www.universalmediaserver.com
INFO 2020-05-12 14:37:41.413 [main]
INFO 2020-05-12 14:37:41.413 [main] Build: 72f327891 (2020-02-23)
INFO 2020-05-12 14:37:41.413 [main] java Java HotSpot(TM) 64-Bit Server VM 1.8.0_241 (64-bit) by Oracle Corporation
INFO 2020-05-12 14:37:41.413 [main] OS: Windows 10 64-bit 10.0
INFO 2020-05-12 14:37:41.413 [main] Maximum JVM Memory: 1.1 GiB
INFO 2020-05-12 14:37:41.421 [main] Language: English (United States)
INFO 2020-05-12 14:37:41.421 [main] Encoding: UTF-8
INFO 2020-05-12 14:37:41.421 [main]
INFO 2020-05-12 14:37:41.421 [main] Working directory: C:\Program Files (x86)\Universal Media Server
INFO 2020-05-12 14:37:41.421 [main] Temporary directory: C:\Users\lach\AppData\Local\Temp\UMS
INFO 2020-05-12 14:37:41.421 [main] Logging configuration file: C:\Program Files (x86)\Universal Media Server\logback.xml
INFO 2020-05-12 14:37:41.421 [main] Logfile: C:\ProgramData\UMS\debug.log
INFO 2020-05-12 14:37:41.421 [main]
INFO 2020-05-12 14:37:41.421 [main] Profile directory: C:\ProgramData\UMS
INFO 2020-05-12 14:37:41.421 [main] Profile directory permissions: drwx
INFO 2020-05-12 14:37:41.421 [main] Profile configuration file: C:\ProgramData\UMS\UMS.conf
INFO 2020-05-12 14:37:41.421 [main] Profile configuration file permissions: -rwx
INFO 2020-05-12 14:37:41.421 [main] Profile name: DESKTOP-FLA0RMU

```

**Logs**

Status: Connected RAM: 56 MB | Bitrate: 0 Mb/s

**Logs**

Universal Media Server 9.3.0

Filter: Off Search log here... Search

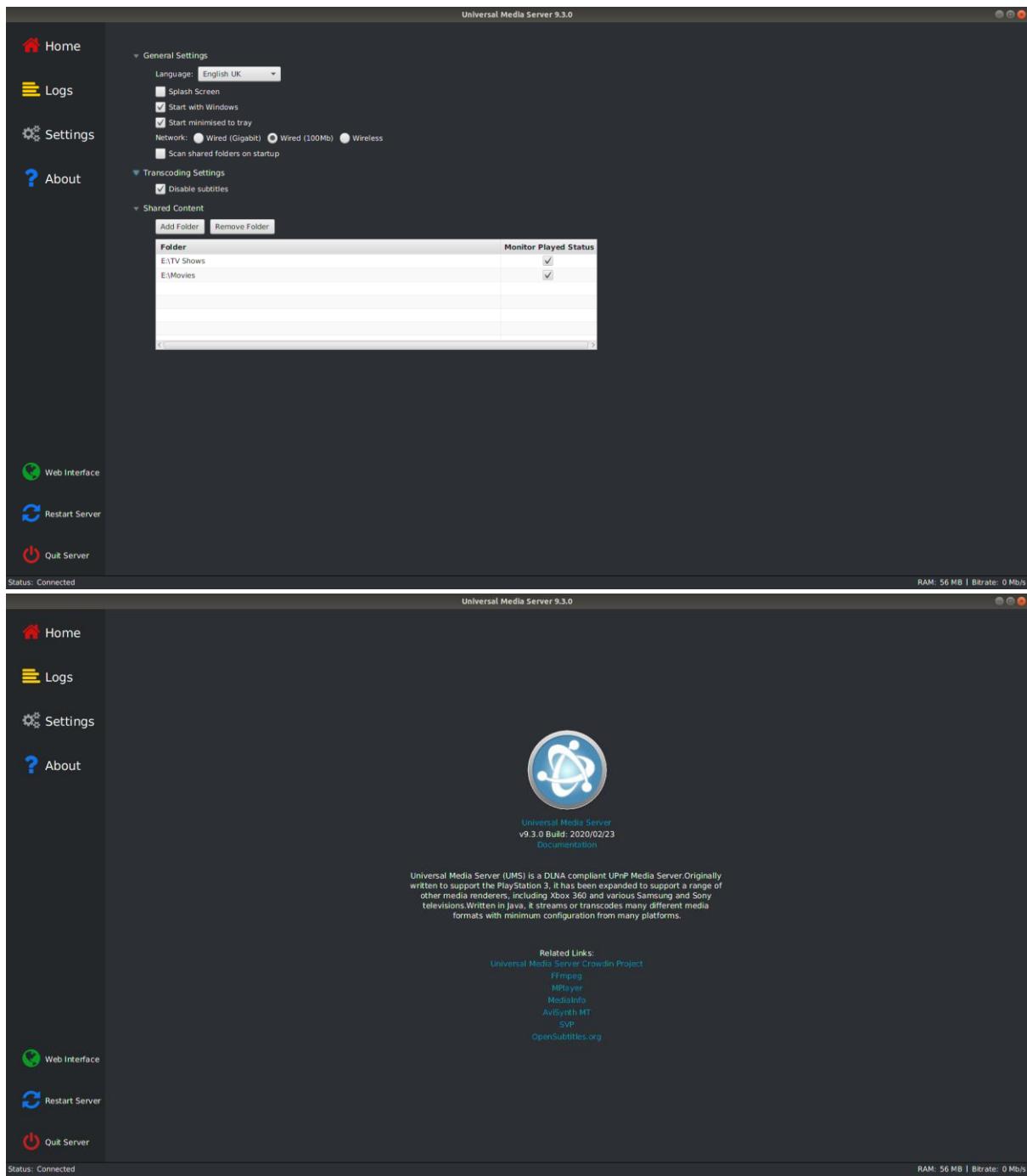
```

DEBUG 2020-05-12 14:37:41.049 [main] Reading log_level: "ALL" (default: "DEBUG")
DEBUG 2020-05-12 14:37:41.049 [main] Default logfile folder set to: C:\ProgramData\UMS\debug
WARN 2020-05-12 14:37:41.049 [main] Could not rename "C:\ProgramData\UMS\debug.log" to "C:\ProgramData\UMS\debug.log.prev"
DEBUG 2020-05-12 14:37:41.103 [main] Logback started with configuration file: C:\Program Files (x86)\Universal Media Server\logback.xml
DEBUG 2020-05-12 14:37:41.103 [main] Reading logging_filter_logs_tab: "ALL" (default: "INFO")
DEBUG 2020-05-12 14:37:41.103 [main] Starting services
DEBUG 2020-05-12 14:37:41.103 [main] Starting ProcessManager
DEBUG 2020-05-12 14:37:41.103 [Process Terminator] ProcessTerminator is starting
TRACE 2020-05-12 14:37:41.103 [Process Terminator] ProcessTerminator is waiting for new tickets
DEBUG 2020-05-12 14:37:41.103 [main] Starting SleepManager
DEBUG 2020-05-12 14:37:41.103 [main] May 12 14:37:41 AWST 2020
DEBUG 2020-05-12 14:37:41.103 [main] Reading cred.path: "C:\ProgramData\UMS\UMS.cred" (default: "")
DEBUG 2020-05-12 14:37:41.296 [main] Writing PID: 10252
TRACE 2020-05-12 14:37:41.296 [System Information Logger] Starting gathering of system information
DEBUG 2020-05-12 14:37:41.413 [main]
INFO 2020-05-12 14:37:41.413 [main] Starting Universal Media Server 9.3.0
INFO 2020-05-12 14:37:41.413 [main] Based on PS3 Media Server by shagrath; copyright 2008-2014
INFO 2020-05-12 14:37:41.413 [main] https://www.universalmediaserver.com
INFO 2020-05-12 14:37:41.413 [main]
INFO 2020-05-12 14:37:41.413 [main] Build: 72f327891 (2020-02-23)
INFO 2020-05-12 14:37:41.413 [main] java Java HotSpot(TM) 64-Bit Server VM 1.8.0_241 (64-bit) by Oracle Corporation
INFO 2020-05-12 14:37:41.413 [main] OS: Windows 10 64-bit 10.0
INFO 2020-05-12 14:37:41.413 [main] Maximum JVM Memory: 1.1 GiB
INFO 2020-05-12 14:37:41.421 [main] Language: English (United States)
INFO 2020-05-12 14:37:41.421 [main] Encoding: UTF-8
INFO 2020-05-12 14:37:41.421 [main]
INFO 2020-05-12 14:37:41.421 [main] Working directory: C:\Program Files (x86)\Universal Media Server
INFO 2020-05-12 14:37:41.421 [main] Temporary directory: C:\Users\lach\AppData\Local\Temp\UMS
INFO 2020-05-12 14:37:41.421 [main] Logging configuration file: C:\Program Files (x86)\Universal Media Server\logback.xml
INFO 2020-05-12 14:37:41.421 [main] Logfile: C:\ProgramData\UMS\debug.log
INFO 2020-05-12 14:37:41.421 [main]
INFO 2020-05-12 14:37:41.421 [main] Profile directory: C:\ProgramData\UMS
INFO 2020-05-12 14:37:41.421 [main] Profile directory permissions: drwx
INFO 2020-05-12 14:37:41.421 [main] Profile configuration file: C:\ProgramData\UMS\UMS.conf
INFO 2020-05-12 14:37:41.421 [main] Profile configuration file permissions: -rwx
INFO 2020-05-12 14:37:41.421 [main] Profile name: DESKTOP-FLA0RMU

```

**Logs**

Status: Connected RAM: 56 MB | Bitrate: 0 Mb/s



### 7.2.3 Testing

Upon review this implementation of a new user interface for our program UMS has successfully met 93% of the specified functional requirements and so far, 90% of the non-functional requirements. This result is satisfactory for this type of project. Down the line, the current implementations can be developed more on to supply the missing functionality and any other requirements that the user may create. Due to ethical and time constraints, user centric testing of the implementation was not feasible.

## 7.3 Alex McLeod

Contributor(s): Alex

### 7.3.1 Implementation Discussion

As seen in my high-fidelity explanations the main goal of my implementation of the universal media server was to create a clean and user friendly interface. This interface allows new users to the media server the ability to easily set up and connect their server, as well as allowing people with a lot of knowledge of media servers the ability to easily edit their server.

#### *User Friendliness*

In terms of user friendliness from the old interface I aimed to improve the ability of the user to easily navigate to different options by grouping together similar screens. E.g. grouping together the different settings tabs under one screen called settings. This would allow the user to find and narrow down their search for an option. E.g. if they wanted to change network settings instead of searching through the different settings tabs, they could easily select the settings option and then select the network tab.

I also added notes to different screens to give new users tips as to how the screen works. For instance, the Media Status screen contains a note stating how to connect new devices and where to go if they are having issues connecting to the server. In the original interface the font size used for different text fields, tabs, buttons and labels is very small making it harder for users to find particular options and navigate the interface. In my design I used a much larger font size and spread out the different elements making sure to use the space available so that users could easily read and distinguish different pieces of text.

One of the screens that I put a lot of work into to increases its user friendliness and convenience was the help screen. In the original universal media server, the help screen is one of the tabs at the top of the screen. It starts with a contents screen where the user can select what topic they would like help with from several links. Upon selecting one of these links they are taken to a new screen. In order to get to the next help screen, they have to scroll to the bottom of the screen and select next. To get to another topic from the contents they are forced to go to the bottom of the screen, select top to go to the contents screen and select the topic they need help with. This is an extremely inconvenient help menu where going between different help topics is cumbersome. To improve this, I first decided to make the help screen its own window allowing the user to read the help information alongside the application itself. I then split the screen into three main sections: the topic pane on the left, the information pane on the right and the next/previous pane on the bottom. The topic pane would allow the user to select the topic they won't help with which would then be displayed in the information pane. This topic pane is scrollable allowing the user to easily scroll to the topic they want. This topic pane adds to the convenience of the screen as the user no longer must go back to the contents screen each time they want to go to a new topic. Instead at any time they can instantly switch to a different topic by selecting it from the topic-pane. The next/previous pane allows the user to easily switch to the next or previous topic no matter where the user is currently in the information pane. Unlike the original interface where the user had to scroll to the bottom of the information screen to go to the next topic.

#### *Visual Appeal*

To increase the visual appeal of the interface I made sure to give it a clean and polished look. To create a clean look, I made sure to spread out different buttons, labels and other elements so they wouldn't be clustered and crammed together. The main example of this can be seen through the buttons at the bottom of the media status, logs, settings, and shared contents screens which I originally had placed right next to each other. This made the screen look less professional as the buttons were so close together. Many professional interfaces like those created by Apple and Microsoft tend to spread different elements out allowing them to be easily distinguished and giving the interface an overall cleaner look. Hence the reason I decided to spread them out.

In the original universal media server under the different settings tabs the different options available were crammed together and placed in one pane making finding an option extremely hard. By using Scrollable panes in my settings tabs, I was able to spread out these different options giving each option its own line. This allows the user to sequentially go through each settings option in that tab rather than having to search a screen where options have been placed everywhere. Similarly, in the old interface the shared content tab had both the shared folders table, web content table and their respective buttons all in one pane causing it to also look too packed together. Hence in my design I spread out these two tables by creating two tabs under shared content one for shared folders and the other for web content. This would allow the user to easily distinguish the information associated with these tables.

To add to the professional look, I also made sure to remove unnecessary borders from elements or decrease their thickness. From my first high-fidelity design I came to realise that thick dividers and borders gave the interface an ugly look, drawing the users attention. I also added colour and icons to each screen as explained in-depth in my prototypes section of the report to add to the visual appeal.

### 7.3.2 User Involvement

Upon starting the application, the Start Menu is displayed to the user. This general start menu like most applications. From here the user can start the actual application by selecting the start application button. This allows the users to read the help information and about information before starting the actual program in the case that they want to learn how to use the application before diving straight in. The user can also select the "start application on start-up" checkbox allowing them to skip this screen on the next start-up adding to convenience. Each of these buttons along with the other buttons in the application glow an aqua colour when hovered over or selected adding to user feedback and overall satisfaction when users are selecting buttons.

Once the start application button is pressed the media status screen is displayed. As stated by the text under Detected Media Renderers the user must turn on media devices on their network for them to be detected. Alternatively, if their device hasn't been detected they can select the Detect button to manually detect any devices on the network. If they want to clear the connected devices, they can select the clear button. If the user hovers over these two buttons a tooltip is displayed giving a better explanation for what they detect and what they clear. Detected devices are displayed under detected Media Renderers and the memory usage displayed below it. If at any time the user wants to close the interface, shutdown the

server, restart the server after a change or open the web interface for the application the user can select one of the four buttons at the bottom. These buttons are shared amongst the four main screens Media Status, Logs, Shared Content and Settings allowing for easy access to these essential commands. These buttons are critical, so each has its own tooltip that is displayed when the user hovers over it stating in more detail what occurs when it is clicked. Such as the close interface button that “closes the interface and adds the application to the system tray”. On the bottom left is the options button. When the arrow is clicked an expanded menu is displayed allowing the user to switch to a different screen or access the help or about windows.

When the user selects “Logs” from the options menu they are brought to the Logs screen. This screen allows the user to check the current logs for their server. It contains a table in the middle displaying all the current logs. Above this are different settings for filtering and a search bar allowing the user to easily search for a log file. Below this are some more advanced settings for creating a TRACE log and packing debugging files. They can also check the full log information by pressing the button at the bottom right.

When the user selects “Shared Content” from the options menu they are taken to the Shared Content screen. On this screen the user can share their personal folders from the computer with the server and web content to the server. This screen starts in the Shared Folders tab as indicated by the highlighted tab stating Shared Folders above the table. This screen displays a table with all the folders the user wants shared with the server from the computer. The table contains a column that states the name of each folder and a column that states whether the folder is being monitored, each row containing its own checkbox. When the checkbox is checked the files within that folder are shared. Below this table are buttons that when clicked the user can add new folders from their system, remove folders that are being shared and scan the currently monitored folders. To remove a folder that is being shared the user must select one of the rows in the table representing a folder causing it to be highlighted aqua and then press the remove button. This remove button also contains a tooltip when hovered over that explains how they first must select one of the folders from the table. From the tabs at the top the user can select the “Web Content” tab to switch to the web content table. This causes the Web content tab to be highlighted. In this table it states the type of content being shared, the folder it comes from and the source. Similarly, content can be added and removed using the buttons at the bottom.

When “Settings” is selected from the options menu the settings screen is displayed. This Screen also uses tabs where the currently selected tab is highlighted aqua. This Screen begins on the General settings screen. This screen contains general settings each setting options being placed in its own label with a description on the left and an editable element on the right allowing the user to change the settings whether it be a checkbox, combobox etc. The user can scroll through these settings as they are within a scroll pane. When the Navigation/Share tab is selected a similar scrollable pane is displayed with settings relating to navigation and sharing. The Transcode tab contains a normal pane where the user can edit checkbox’s, textFields and combo boxes. The user can also select different sections of an accordion element to change video file engine settings. Each section relating to a different engine which when expanded allows the user to edit its settings. The Network pane contains a singular pane with different text fields and elements that can be edited.

When the user selects the “Help” option from the options menu a new help window is opened. The user can then select the particular topic they want help with from the pane on the left which will be displayed on the right. The currently selected tab is highlighted aqua allowing the user to know which is selected. When a user has finished reading a particular section, they can go to the next topic by selecting the next button at the bottom or go to the previous section using the previous button.

Lastly When the “About” option is selected from the options menu the new About window is displayed. This is a simple screen which the user can scroll through to read information about the universal media server.

### 7.3.3 Test Cases

Test Cases:

#### *1. Detect Media Devices for the server*

Steps:

- Select “Start Application” from Start Menu.
- Select the Detect button from the Media Status Screen.

Result: Test case passed as media devices are displayed under detected media Renderer’s (dummy data used).

#### *2. Clear Detected Media Devices*

Steps:

- Select “Start Application” from Start Menu.
- Detect Devices by selecting the detect button from the Media Status screen.
- Select Clear button from Media Status Screen.

Result: Test case passed as media devices under detected media renderers are removed (dummy data removed).

#### *3. Close Interface*

Steps:

- Select “Start Application” from start menu
- From any of the screens Media Status, Logs, Shared Content or Settings press the Close interface button

Results: Passed as the interface closes

#### *4. Shutdown Server*

Steps:

- Select “Start Application” from start menu
- From any of the screens Media Status, Logs, Shared Content or Settings press the Shutdown Server button

Results: Interface closes and a message is displayed stating that the server has been shut down acting as a dummy action as no actual server connected

#### *5. Restart Server*

Steps:

- Select “Start Application” from start menu
- From any of the screens Media Status, Logs, Shared Content or Settings press the Restart button

Results: Message displayed saying that the server has been restarted. Acts as a dummy action as a server is not connected hence nothing is actually restarted.

#### *6. Open Web Interface*

Steps:

- Select “Start Application” from start menu
- From any of the screens Media Status, Logs, Shared Content or Settings press the Open Web Interface button

Results: Message displayed saying web browser has been opened. Acts as a dummy action as there is no backend to open a web browser.

#### *7. Switching Between Screens from options menu*

Steps:

- Select “Start Application” from the start menu
- From any of the screens Media Status, Logs, Shared Content or Settings select the arrow next to the options label to open the expandable menu.
- Select a different screen to display.

Results: Successfully passed as from each screen it is possible to open another screen or separate window if help or about options are selected.

#### *8. Searching for a log file*

Steps:

- Select “Start Application” from the start menu
- Select the Logs screen from the expandable options menu at the bottom left.
- Type the word Yamaha into the text field next to the search button
- Select the search button to search

Results: Successfully passed because in the table only the two logs containing the words Yamaha are displayed.

#### 9. Add folders to be shared to the Shared Folders Table

Steps:

- Select “Start Application” from the start menu
- Select the Shared Content screen from the expandable options menu at the bottom left.
- Press the add folder button at the bottom of the screen

Results: Dummy data is added to the table.

#### 10. Remove Folders from Shared Folders Table

Steps:

- Select “Start Application” from the start menu
- Select the Shared Content screen from the expandable options menu at the bottom left.
- Select the row from the table corresponding to the folder you no longer want to share with the server
- Press the remove folder button at the bottom of the screen

Results: Successful as the folder is removed from the table.

#### 11. Switch between topics in the side scroll pane within the help Screen

Steps:

- Select “Help” from the start menu
- Select the topic you want to read about in the left scroll pane (e.g. installation)

Results: Successful in the case that the installation topic is selected the right information pane successfully switches to information about installation.

12. Go to next or previous topic within the Help screen

Steps:

- Select “Help” from the start menu
- Select the next button or previous button

Results: Successful as when the next button is selected it moves to the next topic in sequential order displaying the correct topic in the information pane on the right until it reaches the last topic and no longer does anything. When the previous button is pressed it successfully goes to the previous topic displaying in the information pane on the right until it is back at the first topic and no longer does anything.

## 7.4 Brendan Scarfone

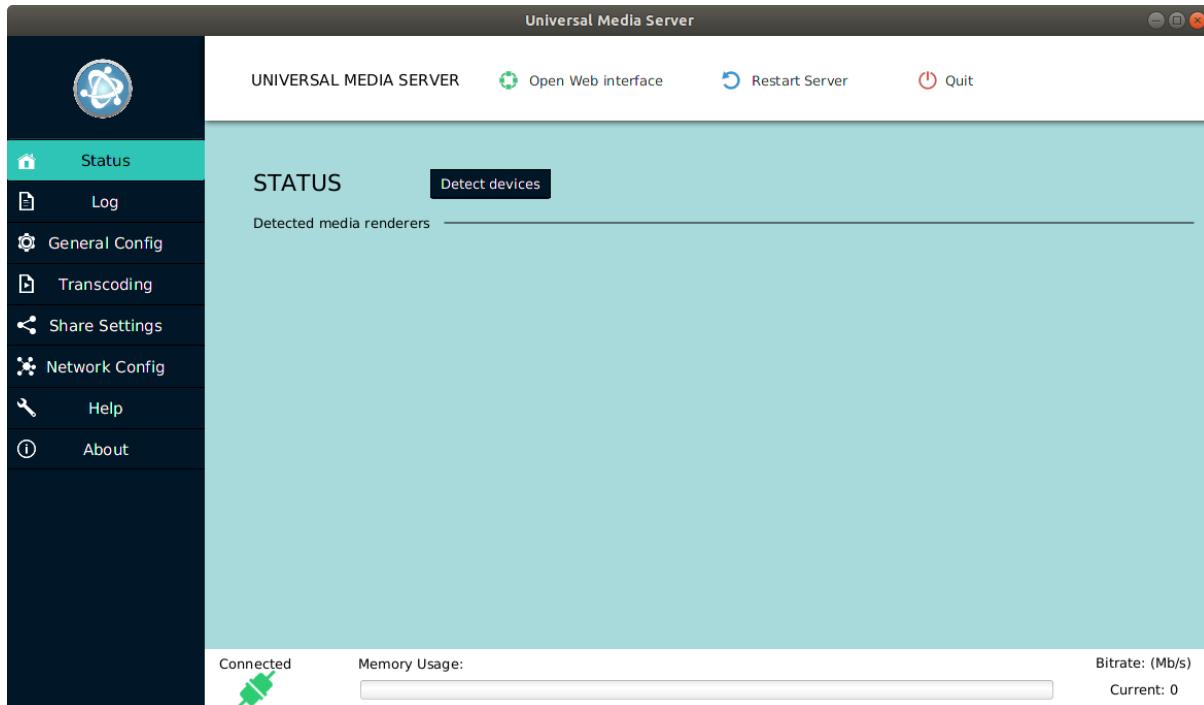
Contributor(s): Brendan

### 7.4.1 Implementation Discussion

With my implementation I focused mostly on improving the attractiveness of the application as well as moving/adding to some of the settings pages as I felt they did not fit in their current placements. Things such as moving the network and http settings out of the general configuration tab into its own tab as well as adjusting colour schemes to something more modern than the default light grey. I believe that the original tabs had decent layout so decided to leave the general idea behind the separate windows. I have added tooltips to many sections and buttons that I felt needed more explanation. The information about these areas can be found in the help tab from the original design but I felt that placing them in a tooltip prevents the user from having to search the applications help tab or the internet to understand the complex settings.

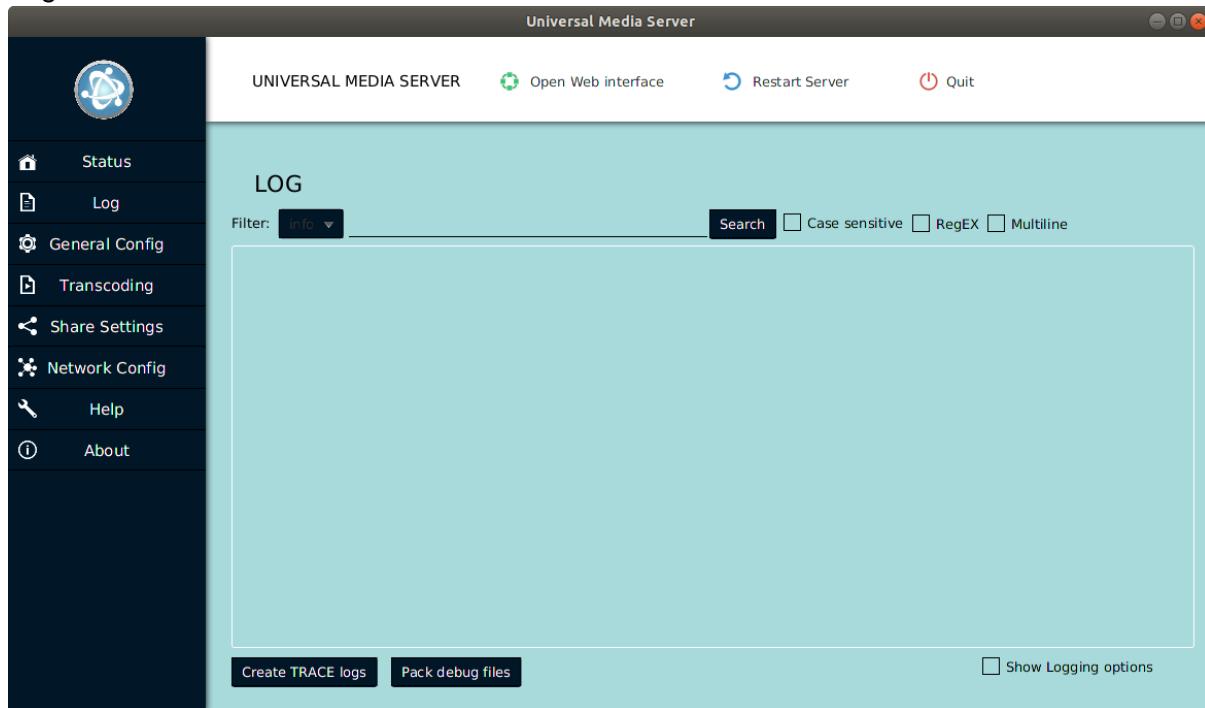
The design takes inspiration for multiple modern sources which include the side navigation section. I feel that the navigation on the side looks better than the original top navigation section.

Status tab:



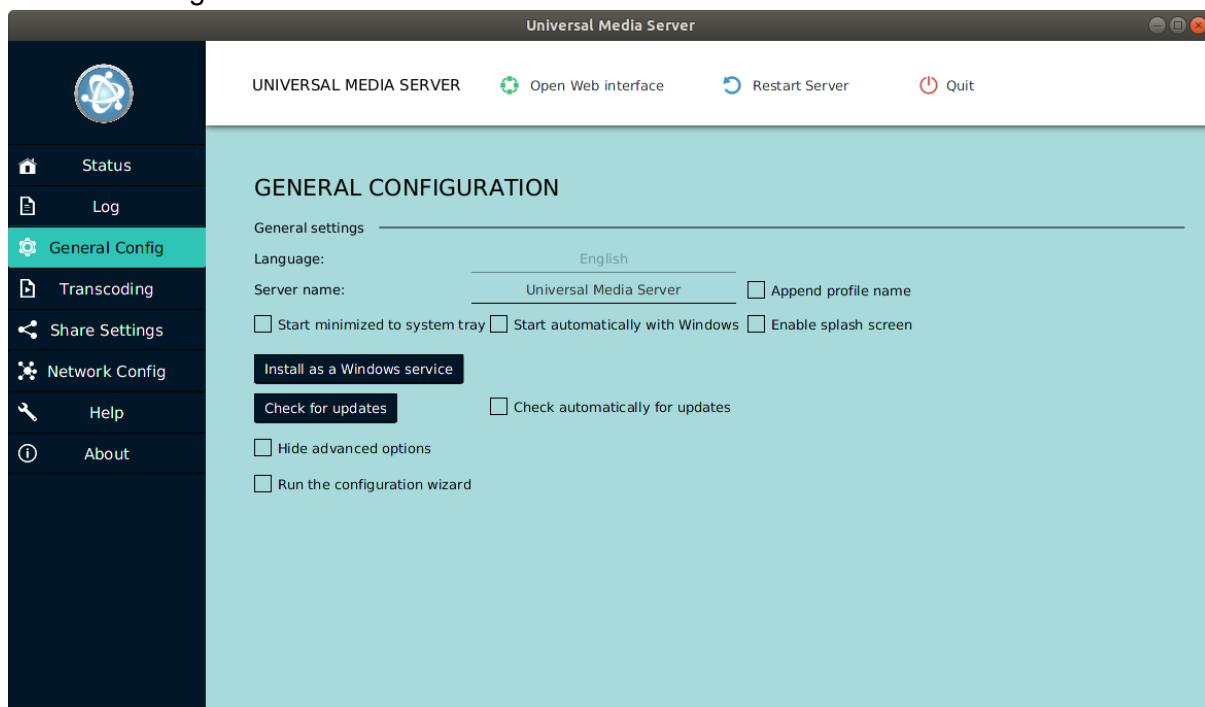
With the final implementation I decided to change the colour scheme after the feedback. I have stuck to a simple dark blue for buttons and for the side navigation section. I have also added updated icons for the Web interface, restart, and quit buttons. As well as adding icons to the side navigation buttons to represent their intended task. The status tab itself has not been changed much except for the different symbols for the connection status, replacing the original that was used as more of a placeholder. The detect devices button has been added to show how actual detection would appear but does not actually work.

## Logs tab:



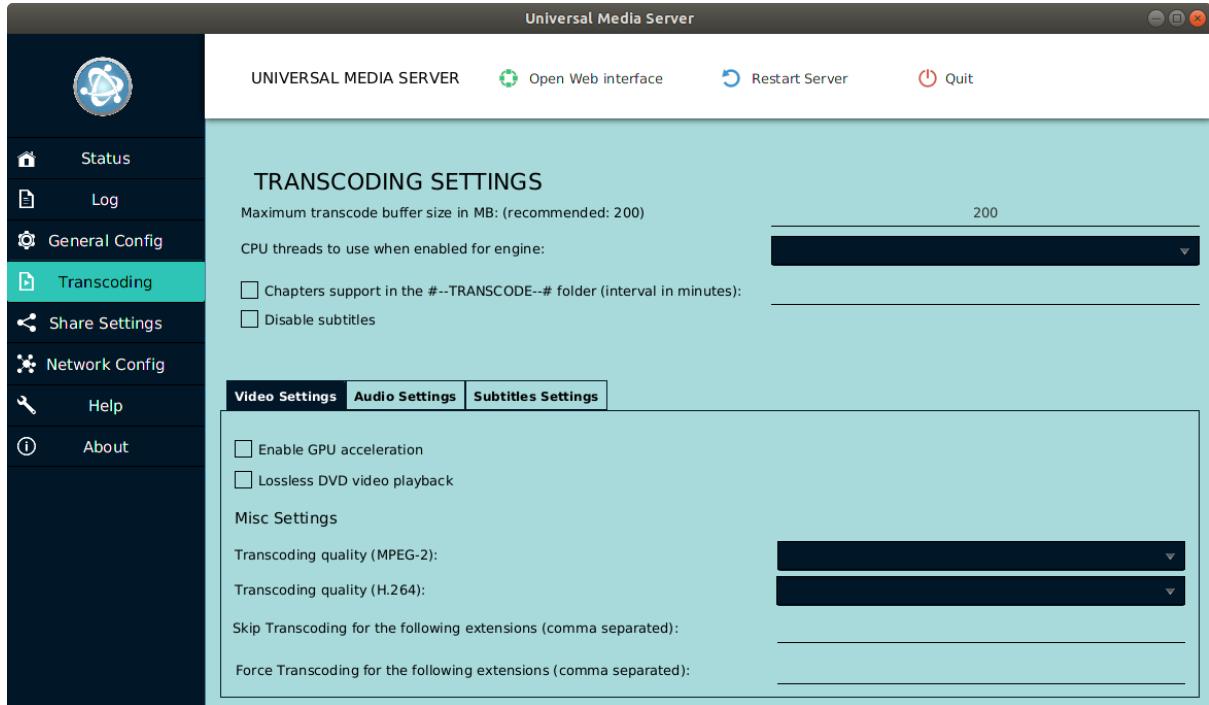
The colour change continues with the log tab, the previous background colour was not a good fit and as such I decided to change all the windows colours to the light blue. The buttons along the sides of the log area have been changed removing the default button style for a more updated colourful button which matches the blue of the navigation section. The checkboxes have also been changed from the default, they are now just using a black border and the same background colour as the window. The log area colour has also been changed from the white the light blue but has been given a white border to separate it from the rest of the window.

## General Config tab:



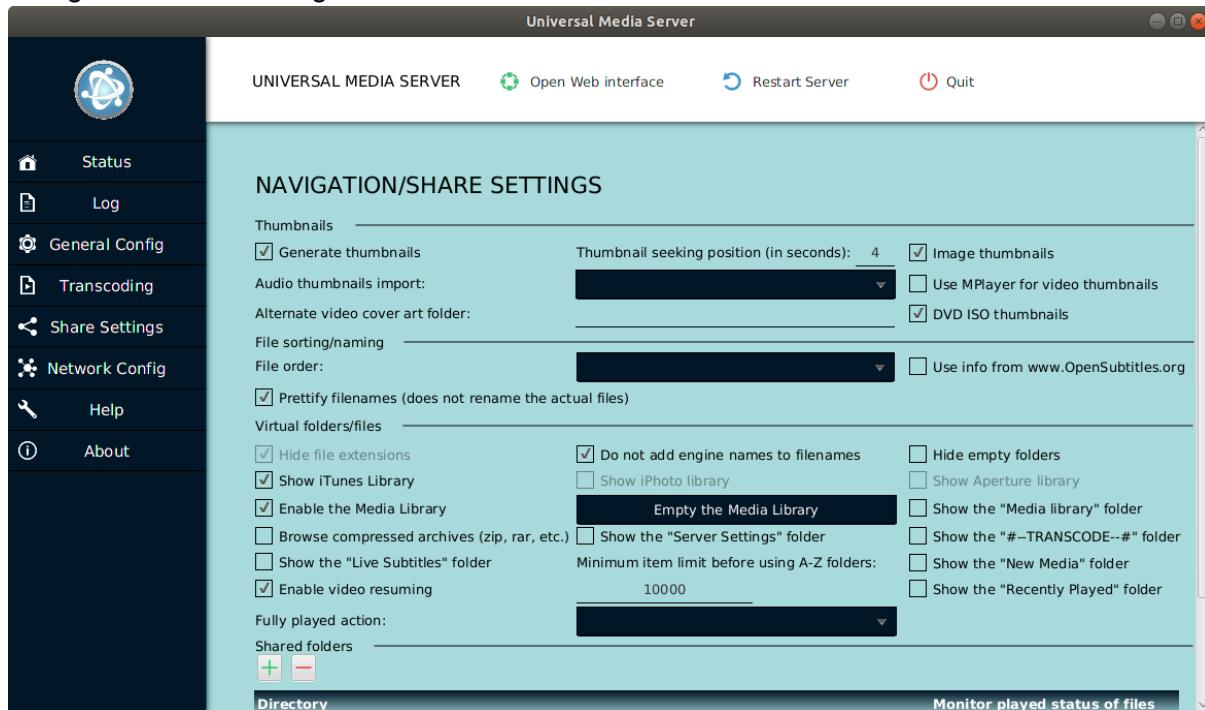
The general configuration tab has also been updated. When selecting a different window, the button on the navigation section is also selected and changed to a lighter blue. The text area style has been changed from the default box to a simple line by removing the background and adding a black border on only the bottom using the custom css. The buttons and checkbox have also been replaced to continue the theme from the other sections.

Transcoding tab:



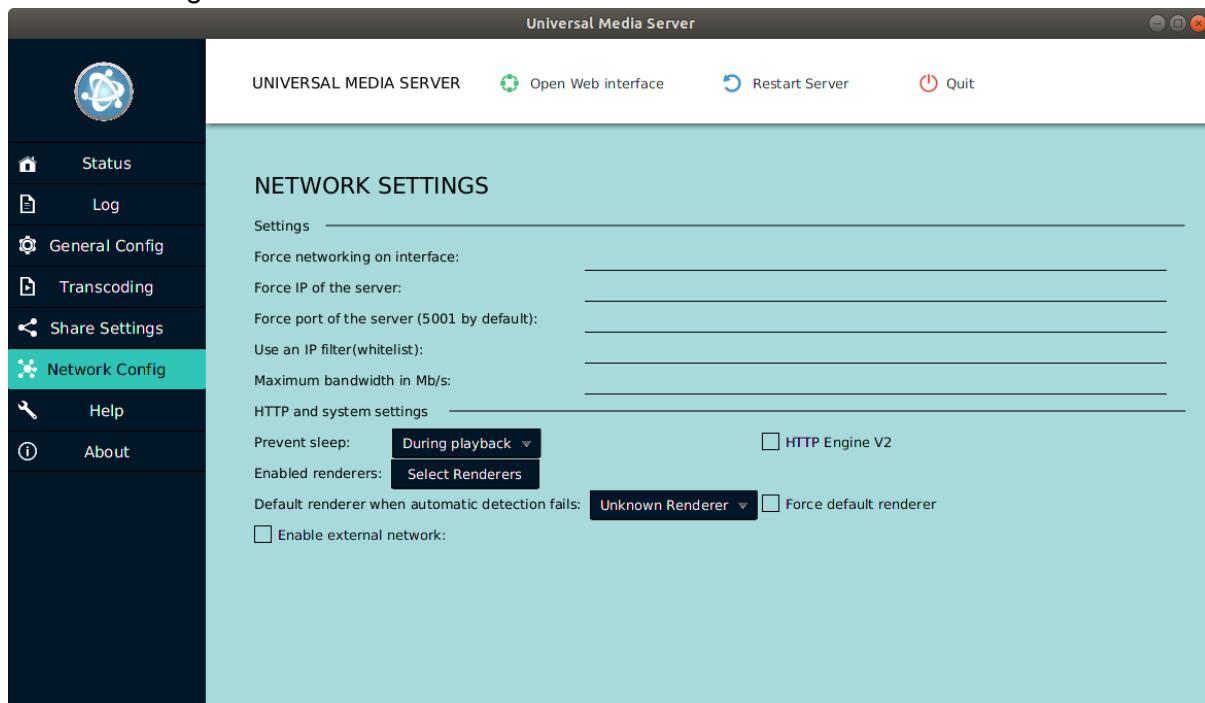
For the final implementation I added the transcoding setting window. This window contains the settings for how media is received by devices. This includes settings for video quality and codecs as well as audio and subtitles settings. I follow the same theme as the previous windows with the light blue background and dark blue buttons. The different settings are separating into their own tabs which changes the small section surrounded in the black border.

## Navigation/Share settings:



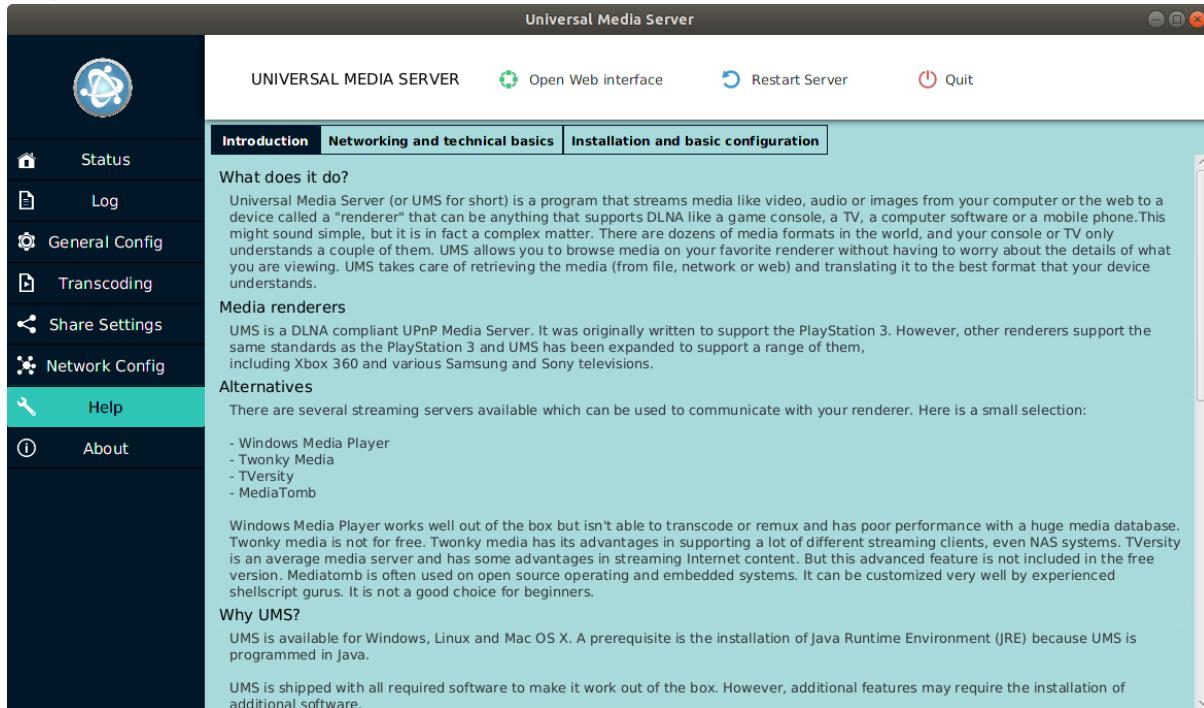
The navigation/share Settings window is also new but continues the same theme from the previous setting pages. This window gives the user the ability to manipulate how the server's media is seen from media devices. Things such as thumbnails and the order of the files. It also contains the shared folders which can be added to or removed.

## Network config tab:



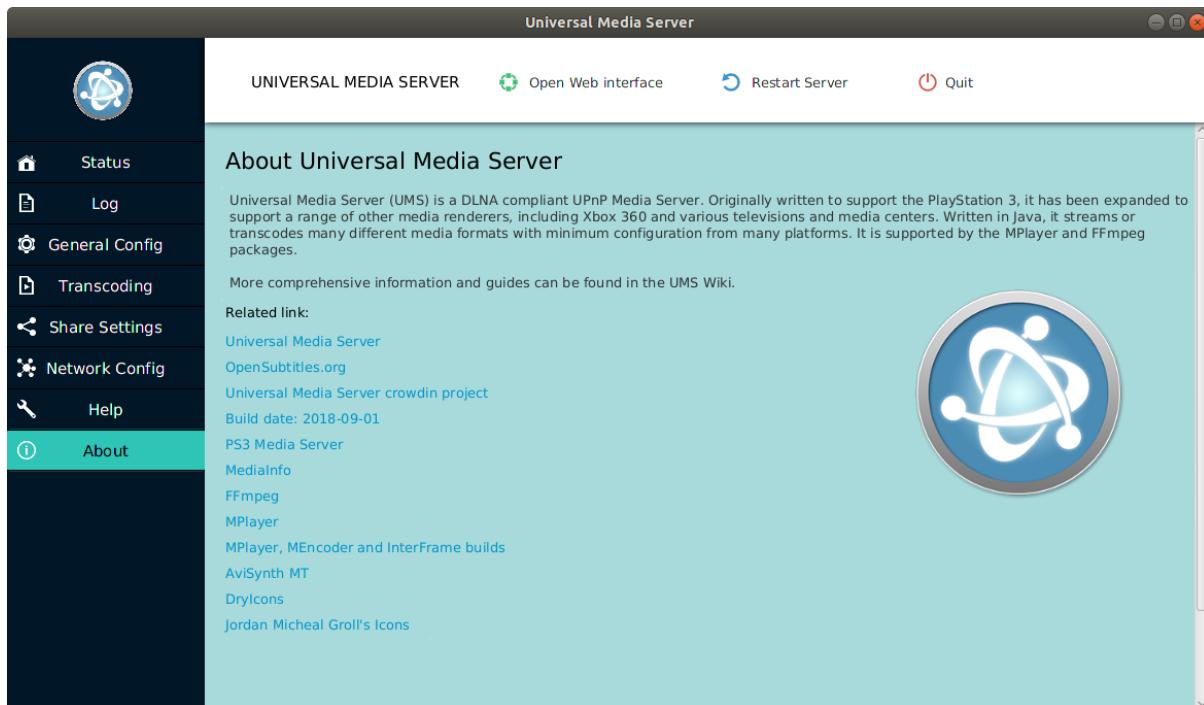
This window is very similar to the prototype network settings. Although the text fields have been added as well as the buttons and checkboxes getting the new theme.

## Help tab:



The help tab is the one of the simpler tabs. This window just contains information about the program as well as the installation, troubleshooting and technical basics. This helps the user with any of the settings or options that are not explained using the tooltips.

## About tab:



The final window for the implementation is the about tab which contains information about UMS. It also includes links to relevant sites and programs which may be used with UMS and or help with understanding the settings and operation of the media server.

#### **7.4.2 User Involvement**

The program opens to the status tab which displays the current devices available from streaming. This tab also shows the current server status as well as its memory usage. From the navigation section on the left the user can easily select the different tabs which will open windows related to the setting and operation of the media server. The user can also detect devices using the button at the top of the status tab.

The log tab is the second available window under the status tab. This window allows the user to view the current debug log to check for errors and warnings inside the program. It also contains settings to search and filter the log. There are also more advanced settings which allow the user to create trace logs and to pack the debug files as well as making the search and filter case sensitive as well as others. The search button does work if the user was to search a String but only returns the first occurrence.

The settings for the application are separated into different windows as there is quite a lot of customization that is available for this program. The first tab being the general config, this allows the user to customize the name of the server and the default language which is set to English when the program is installed. This tab also allows the user to customize how UMS operates on their device with options such as starting automatically when the computer is turned on. The next tab is the transcoding settings, this allows users to customize how the application transmits media to devices on the network. It allows users to choose if subtitles are included and what video and audio codecs they want to use. It also contains general transcoding settings such as how many cores to use for operation and the buffer size. Share settings is next. This window allows the user to customize how the device views the media server and what folders are being shared on the server. It allows the user to have thumbnails for easy identification of video files as well as choosing how the media folders are viewed on devices. The user can also add and remove shared files from this window. The last settings page is the network config. This section was originally part of the general settings, but I felt it made more sense to put it in its own tab. Network settings allow the user to customize the IP and ports of not only the media server but also the http server which is started by the application.

The second last tab is the help tab. This window gives the user an introduction to the application as well as how to set up the media server for your own personal needs. It contains information about installing the application as well as basic settings and configuration to get UMS working. In the original design it also contained tabs which explained what settings do on the different settings windows this has been changed to tooltips as they are easier than digging around in the help section.

The final tab is the about tab. This window gives the user the basic understanding of what UMS is and how it works. It also links to many different websites that contain relevant information or applications which may be useful for some users. All of these hyperlinks do open the directed website. The links have also been fixed from the original as some linked to disabled websites

### **7.4.3 Test Cases**

**8. Detect media devices**

- a. Run the program
- b. Press the detect devices button on the status tab

The detect media devices button will show how actual media devices would appear on the program (uses dummy code doesn't actually find devices)

**9. Changing windows**

- a. Run the program
- b. From any window press any button on the navigation section to the left

All navigation buttons will open the requested tab

**10. Quit the application**

- a. From any window press the quit button at the top

The quit button will exit the program in the same way as the x at the top

**11. The user wants to find subtitles for a file**

- a. Press the about tab
- b. Locate and press the OpenSubtitles.org link

All hyperlinks on the about page should work with the default browser on the system

**12. The user wants to add a file to the shared files**

- a. Press the Share settings button
- b. Press the + button above the directory

The program will add dummy values to simulate how UMS would add them

**13. The user wants to remove a file from the shared files**

- a. Press the Share settings button
- b. Select the value to remove
- c. Press the - button

The program will remove any selected dummy values to simulate how UMS would remove them

**14. The user wants to search the log**

- a. Press the log button
- b. Enters the String they want to search
- c. The users presses the search button

The search button will find the first occurrence of the word searched

# 8.0 Evaluation

## 8.1 Meetings

Contributor(s): Lachlan

Throughout this project, meetings were held every 3 to 4 weeks on a Thursday. The first meeting was held in person during or after the tutorial sessions. The other meetings were held via a voice call in a Discord server in response to the COVID-19 pandemic. During these meetings some milestones were set that are to be completed by the next meeting. For more general communication, the text channels in the Discord server were used for quicker responses.

## 8.2 Milestones

Contributor(s): Lachlan

Date	Tasks To Be Completed
13th Mar	<ul style="list-style-type: none"><li>• Assign a role to every group member</li><li>• Choose a suitable program for developing a new GUI</li><li>• Write up and submit the program proposal.</li></ul>
20th Apr	<ul style="list-style-type: none"><li>• Requirements elicitation with survey</li><li>• Produce a list of requirements.</li><li>• Analyse competitor programs</li><li>• Develop low-fidelity prototype of GUI</li><li>• Produce use cases/user stories</li><li>• Develop high-fidelity prototype of GUI</li><li>• Submit prototypes to client for feedback</li><li>• Have a list of questions ready for the client meeting in regard to their feedback</li></ul>
27th Apr	<ul style="list-style-type: none"><li>• Take the client's feedback into account for the implementation</li><li>• Make significant progress on the implementation</li></ul>
18th May	<ul style="list-style-type: none"><li>• Implementations are completed</li><li>• Report is completed</li><li>• Submit report</li></ul>

## 8.3 Risk Management

Contributor(s): Lachlan

Throughout this project there were many risks to be considered. However, the main risks that were associated with the success of the project are;

### The client rejects the prototype

The risk involving the client not liking major parts of a designed prototype would result in a longer implementation time trying to correct the parts of the design to how the client wants. To mitigate this risk, the client meeting for feedback on the design should be held as soon as possible to provide the team or member with the needed extra time to fulfil the client's feedback.

### The technical feasibility of the prototypes and requirements

The team members have limited skills in regard to GUI development and won't be sure on what is technically feasible for a GUI and what isn't. The team could have specified a requirement without knowing it could be highly difficult or impossible to implement without being a massive time sink. To mitigate this risk, each requirement the team had gathered was researched to prove that it was possible to implement in our final designs.

## **8.4 Problems Encountered**

Contributor(s): Lachlan, Brendan

The main problem the team encountered was the COVID-19 situation with it affecting the face to face meetings. As the team was no longer able to perform the meetings in person at university, it was decided to move them over to a Discord server made by one of the members. This Discord server was then used for all future communication.

A major problem I faced (Brendan) was the SceneBuilder that I used to make most of the project. It repeatedly crashed on me causing me to lose lots of progress during the creation of the GUI. Also, the VM provided did crash on me quite often as well. This may only apply to me, but it was a major nuisance. I (Lachlan) can back this up as well, NetBeans crashing and the VM crashing were such an annoyance that I ended up developing it on my Windows machine. However, that brought up its own problems with Linux compatibility in regard to opening browsers and file viewers which I had to spend extra time of my own to fix.

## **8.5 Conclusion**

Contributor(s): Lachlan

Using an agile methodology and focusing on client feedback allowed our team to produce four low-fidelity and high-fidelity prototypes for the Universal Media Server, a program used to share content within the user's local network. These prototypes followed the set functional requirements that were gathered from existing users and competitor programs. The implementations of these prototypes were then developed using JavaFX into fully functional graphical user interfaces. The testing of the implementations against the requirements of the interface ensured that they were met.

## **9.0 References**

Serviio (2020). Serviio Media Streaming Server. Retrieved from <https://serviio.org/>

Plex (2020). Plex, Stream Smarter. Retrieved from <https://www.plex.tv/en-au/>

# **10.0 Appendix**

## **10.1 Initial User Survey Responses**

Contributor(s): Alex McLeod

Universal Media Server User Survey:

Note: For current users of the Universal Media Server application.

Click the answer to each question that relates to you the most.

1.What age are you?

- a) 10 – 19 years old
- b) 20 – 39 years old
- c) 40 – 49 years old
- d) 50 years old or older

2.What is your occupation?

- a) Primary school/high school student
- b) University student
- c) Computing student
- d) Full time worker
- e) Involved in full time work related to computing

3. On average how many hours would you spend watching videos in a day?

- a) Less than 1 hour
- b) 1 – 3 hours
- c) More than 3 hours

4.Would you say that you have a good understanding as to how media servers work?

- a) Yes
- b) No

5. Did you seek help from another individual to get your media server working?

- a) Yes

b) No

6. Are you able to easily find the settings you want to change?

a) Yes

b) No

7. Do you believe that the look of the application is appealing?

a) Yes

b) No

8. Is the application satisfying to use?

a) Yes

b) No

## 10.2 Meeting Minutes

### 10.2.1 05/03/2020

Contributor(s): Lachlan

Meeting Minutes  
HCI Assignment 1  
05/03/2020

#### Attending:

Aidan Leavy  
Lachlan Mackenzie  
Alex McLeod  
Brendan Scarfone

#### Absent:

None

#### Items:

1. For the role assignments, don't need Psychologist, Data Analyst, Usability Engineer.  
Have 1 Project Manager, 2 Graphic Designers / Programmers and 1 Document Controller
2. Proposed program should be open source and written in Java, this will make implementation for bonus marks easier
3. Setup a Discord server for quick communication

#### Milestones To Be Completed By 13/03/2020:

1. Assign a role to every group member
2. Choose a suitable program for developing a new GUI
3. Write up and submit the program proposal.

## **10.2.2 26/03/2020**

Contributor(s): Lachlan

### Meeting Minutes HCI Assignment 1 26/03/2020

#### **Attending:**

Aidan Leavy  
Lachlan Mackenzie  
Alex McLeod  
Brendan Scarfone

#### **Absent:**

None

#### **Items:**

1. Look at other similar programs to see what functionality they have
2. Start on the list of requirements for the new user interfaces
3. Produce a survey
4. Do a quick sketch of any ideas for a GUI

#### **Milestones To Be Completed By 20/04/2020:**

1. Requirements elicitation with survey
2. Produce a list of requirements.
3. Analyse competitor programs
4. Develop low-fidelity prototype of GUI
5. Produce use cases/user stories
6. Develop high-fidelity prototype of GUI
7. Submit prototypes to client for feedback
8. Have a list of questions ready for the client meeting in regard to their feedback

### **10.2.3 16/04/2020**

Contributor(s): Lachlan

### Meeting Minutes HCI Assignment 1 16/04/2020

#### **Attending:**

Aidan Leavy  
Lachlan Mackenzie  
Alex McLeod  
Brendan Scarfone

#### **Absent:**

None

#### **Items:**

1. Finish up our high-fidelity prototypes ready for submission
2. Complete the last couple use cases and stories

#### **Milestones To Be Completed By 20/04/2020:**

1. Requirements elicitation with survey
2. Produce a list of requirements.
3. Analyse competitor programs
4. Develop low-fidelity prototype of GUI
5. Produce use cases/user stories
6. Develop high-fidelity prototype of GUI
7. Submit prototypes to client for feedback
8. Have a list of questions ready for the client meeting in regard to their feedback

## **10.2.4 14/05/2020**

Contributor(s): Lachlan

### Meeting Minutes HCI Assignment 1 14/05/2020

#### **Attending:**

Aidan Leavy  
Lachlan Mackenzie  
Alex McLeod  
Brendan Scarfone

#### **Absent:**

None

#### **Items:**

1. Make sure we are all almost finished on our implementations
2. Work on the individual sections of the report
3. Got to do the evaluation part as well, include COVID-19 as a problem

#### **Milestones To Be Completed By 18/05/2020:**

1. Implementations are completed
2. Report is completed
3. Submit report