JASLiN

Alan Lovell

3 May 2020

A project report submitted in partial fulfilment for the degree of

**Bachelor of Science in Computing**

**School of Physical Sciences and Computing**

**University of Central Lancashire**

Abstract

**Problem:**

Problem: What you tackled, and why this needed a solution

**Objectives:**

Objectives: What you set out to achieve, and how this addressed the problem

**Methodology:**

Methodology: How you went about solving the problem

**Achievements:**

Achievements: What you managed to achieve, and how far it meets your objectives.

Attestation

I understand the nature of plagiarism, and I am aware of the University’s

policy on this.

I certify that this document reports original work by me during my University project.

**Signature** **Date**

Acknowledgements

Acknowledge anyone who has helped you in your work such as your supervisor, technical support staff, fellow students or external organisations. Acknowledge the source of any work that is not your own.

Table of Contents

[Abstract i](#_Toc37594145)

[Attestation ii](#_Toc37594146)

[Acknowledgements iii](#_Toc37594147)

[Table of Contents iv](#_Toc37594148)

[List of Figures vi](#_Toc37594149)

[List of Tables vii](#_Toc37594150)

[List of Listings viii](#_Toc37594151)

[1 Introduction 1](#_Toc37594152)

[2 Background and Related Work 2](#_Toc37594153)

[3 Project Planning 3](#_Toc37594154)

[3.1 Introduction 3](#_Toc37594155)

[3.2 Methodology 3](#_Toc37594156)

[3.3 Requirements 3](#_Toc37594157)

[3.4 Potential Solutions 3](#_Toc37594158)

[3.5 Tools and Techniques 3](#_Toc37594159)

[3.6 Legal, Social and Ethical Issues 3](#_Toc37594160)

[3.6.1 First Subsection 3](#_Toc37594161)

[3.6.2 Second Subsection 3](#_Toc37594162)

[3.7 Another Section 3](#_Toc37594163)

[3.7.1 Summary 4](#_Toc37594164)

[4 Design 5](#_Toc37594165)

[4.1 Introduction 5](#_Toc37594166)

[4.2 System Design 5](#_Toc37594167)

[4.3 User Interface Design 5](#_Toc37594168)

[4.4 Summary 5](#_Toc37594169)

[5 Implementation 6](#_Toc37594170)

[5.1 Introduction 6](#_Toc37594171)

[5.2 Section 6](#_Toc37594172)

[5.2.1 Subsection 1 6](#_Toc37594173)

[5.2.2 Subsection 2 7](#_Toc37594174)

[5.3 Section 7](#_Toc37594175)

[5.4 Summary 7](#_Toc37594176)

[6 Test Strategy and Evaluation 8](#_Toc37594177)

[6.1 Introduction 8](#_Toc37594178)

[6.2 Functional Testing 8](#_Toc37594179)

[6.3 Non-Functional Testing 8](#_Toc37594180)

[6.4 User Testing 8](#_Toc37594181)

[6.5 Evaluation 8](#_Toc37594182)

[6.6 Summary 8](#_Toc37594183)

[7 Evaluation, Conclusions and Future Work 9](#_Toc37594184)

[7.1 Project Objectives 9](#_Toc37594185)

[7.2 Self-Evaluation 9](#_Toc37594186)

[7.3 Project Evaluation 9](#_Toc37594187)

[7.4 Applicability of Findings to the Commercial World 9](#_Toc37594188)

[7.5 Conclusions 9](#_Toc37594189)

[7.6 Future Work 10](#_Toc37594190)

[References 11](#_Toc37594191)

[Appendix 1 – Project Proposal 13](#_Toc37594192)

[Appendix 2 – Technical Plan 14](#_Toc37594193)

[Appendix 3 – Title of Appendix 15](#_Toc37594194)

List of Figures

[Figure 1 - Highly Technical Diagram 3](#_Toc476373580)

List of Tables

[Table 1 - Test Results 6](#_Toc476373581)

List of Listings

This list only applies to you if you use code snippets in your report. If you don’t have any listings, remove this whole section including the heading ‘List of Listings’.

[Listing 1 - [Main.java] The main class of the program 5](#_Toc476373582)

# Introduction

For editorial consistency, it is important to use Word styles properly. Word 2003 onwards has so-called ‘quick styles’. If the styles referred to below are not visible on the Home ribbon in the Styles category, choose ‘Apply Styles’ from the down arrow at the bottom right of the Styles category. Styles can then be applied from the drop-down box. To make a style visible as a quick style, choose Apply Styles, then click Styles (the AA icon) or use ‘Alt + Ctrl + Shift + S’, then right-click on the style and then ‘Add to Style Gallery’.

Chapters are entered using the ‘Heading 1’ paragraph style. The Heading 1 style automatically moves to the start of a new page and supplies the next chapter number. Pressing enter on a ‘Heading 1’ heading automatically inserts a ‘Heading 2’ heading underneath.

There should not be any text between a parent heading and its first sub-heading. For example, when you want to write an introductory section for the following sections, give that introductory section an own ‘Introduction’ heading instead of writing it between the heading preceding these sections and the first sub-heading.

As an example: This comment text is incorrectly placed between a ‘Heading 1’ (‘Introduction’) and a ‘Heading 2’ (‘Background and Context’).

Most text uses the ‘Normal Project Body’ paragraph style with 10-point Arial, 1.5-line spacing, single-sided pages.

In general, use the default spacing that headings and paragraphs give you. Avoid using new-lines or spaces to format text. If you need to use quotes, preferably use single curly quotes ‘…’. If you wish to emphasise something, use the ‘Emphasis’ style. In addition, also a ‘**Strong’** style is preconfigured.

Remember to Save frequently while you are working! Check that AutoSaving is enabled under options -> save -> 'Save AutoRecover information every 5 minutes'.

# Background and Related Work

Copy and paste your background and related work submission here. MAKE SURE IT MATCHES THE FORMATTING FOR THIS TEMPLATE. You may need to make some minor changes to make it look correct. Do not list your references at the end of this chapter, list them in the REFERENCES section at the end of this report. You must amalgamate all your references together, listing them in alphabetical order by author surname.

# Project Planning

## Introduction

Each of your chapters should have an introduction to tell your readers what they will find in the chapter.

## Methodology

## Requirements

## Potential Solutions

## Tools and Techniques

## Legal, Social and Ethical Issues

### First Subsection

If necessary, also use subsections. Subsections are entered using the ‘Heading 3’ paragraph style (all these heading styles are self-numbering). Do not go lower than Heading 3. Try to avoid if possible.

### Second Subsection

## Another Section

As an example of a figure, consider Figure 1.

To place a figure, insert the picture/diagram/etc. where you want it to be, make sure it is selected and then apply the ‘Project Figure Title’ style which centres the figure horizontally.

Captions are entered through the ribbon menu under ‘References’ -> ‘Insert Caption’ or through right-clicking an image and selecting ‘Insert Caption’. Add the caption text in the box, separated with a dash as the example below shows.

Each figure is numbered automatically, and it is possible to make cross-references to figures.

diss-fig

Figure 1 - Highly Technical Diagram

### Summary

Write a short summary at the end of each chapter. Do not use the words ‘In summary’, we know what it is from the title.

# Design

## Introduction

Each of your chapters should have an introduction to tell your readers what they will find in the chapter. Remember to change the sub chapter headings to some that are suitable to your project.

## System Design

Text goes here.

## User Interface Design

Text goes here.

## Summary

Write a short summary at the end of each chapter. Do not use the words ‘In summary’, we know what it is from the title.

# Implementation

## Introduction

Each of your chapters should have an introduction to tell your readers what they will find in the chapter. Depending on what type of project you are doing, you may name this chapter ‘Investigation’.

## Section

### Subsection 1

Code can be formatted using the ‘Code’ style. An example is shown below. It can be a little bit tricky to keep the formatting when pasting from an IDE but the following works for most IDEs: Copy the text from the IDE, paste it in Word, select the pasted code and change the style to ‘code’. It is worth noting that spell checking is deactivated for the ‘Code’ style.

public class Main {  
  
 public static void main(String[] args) {  
 System.*out*.println("Hello World!");  
 }  
}

Listing 1 - [Main.java] The main class of the program

Captions are entered through the ribbon menu under ‘References’ -> ‘Insert Caption’. Select ‘Listing’ (or add a new Label called ‘Listing’ if it does not already exist) and add the caption text in the white box, separated with a dash as the example above shows. Think about a naming convention for listings and stick to it throughout the report. For example, as seen above,  
‘[ClassName or Filename] Description’.

In case you are mixing multiple programming languages: Consider stating the language name in the caption if it is not obvious from the file name or when there is no file name to refer to. For example, when you use XML and HTML, JavaScript and TypeScript or other languages with similar syntax. A suggestion might be to add the language in parenthesis at the end.

It is also possible to use the ‘code’ style “inline” to highlight commands in normal text by selecting the words to highlight and choosing the ‘code’ style. For example:

This example demonstrates the ping 127.0.0.1 command.

Make sure to write the whole text first and select the part you want to highlight afterwards. When there is no selection, Word applies the selected style to the whole paragraph.

### Subsection 2

## Section

## Summary

Write a short summary at the end of each chapter. Do not use the words ‘In summary’, we know what it is from the title.

# Test Strategy and Evaluation

## Introduction

Each of your chapters should have an introduction to tell your readers what they will find in the chapter. In this chapter you should introduce your test strategy – how have you tested your artefact. You should also talk about user testing. How did you test with real people? How did you select them? What did you ask them to do? What ethical considerations did you adhere to? In this chapter you will also discuss how you have carried out an evaluation of your artefact. This is not the same thing as a total project evaluation.

## Functional Testing

To add a caption to a table, either select the whole table (e.g. by clicking on the + symbol in the upper left corner of the table), right-click it and choose ‘Insert Caption’ or click in any table cell and select ‘References’ -> ‘Insert Caption’ from the ribbon menu. Choose ‘Table’ as label and ‘above the item’ as position. Add the caption text in the box, separated with a dash as the example below shows.

Table 1 - Test Results

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Non-Functional Testing

Text goes here.

## User Testing

Text goes here.

## Evaluation

Text goes here.

## Summary

Write a short summary at the end of each chapter. Do not use the words ‘In summary’, we know what it is from the title.

# Evaluation, Conclusions and Future Work

## Project Objectives

Summarise what you have achieved.

## Self-Evaluation

This section is about yourself. Be honest. Look at where you were situated at the beginning of the project and where you are now. What have you learnt on a personal level, what have you found out about yourself? Try to reflect upon individual goals, experiences, and incidents. No one is perfect, and it is very likely that you will recall both good and bad experiences.

The purpose of the evaluation process is to highlight strengths, correct performance weaknesses, and develop unused skills and abilities. To do this, you must be willing to recognise areas that need improvement or development.

## Project Evaluation

Stand back and evaluate what you have achieved and how well you have met the objectives. Evaluate your achievements against your objectives in section 1.2. Demonstrate that you have tackled the project in a professional manner.

(The previous paragraph demonstrates the use of automatic cross-references: The ‘1.2’ is a Cross-reference to the text in a numbered item of the document, it is not literal text but a field. The number that appears here will change automatically if the number on the referred-to section is altered, for example if a chapter or section is added or deleted before it. Cross-references are entered using Word's **Insert** or **References** menu. Cross-references are set to update automatically when printed but may not do so on-screen beforehand; you can update a field manually on-screen by right-clicking on it and selecting Update field from the pop-up menu or by selecting the whole document and pressing F9.)

## Applicability of Findings to the Commercial World

Summarise what you have achieved and how it can apply to the commercial world.

## Conclusions

Summarise what you have achieved. Do not use the words ‘In conclusion’ or ‘to conclude’ or any derivative of those. We know this is the conclusions from the title.

## Future Work

Explain any limitations in your results and how things might be improved. Discuss how your work might be developed further. Reflect on your results in isolation and in relation to what others have achieved in the same field. This self-analysis is particularly important. You should give a critical evaluation of what went well, and what might be improved.

References

Cuevas, R. et al. (2010a) 'A collaborative P2P scheme for NAT Traversal server discovery based on topological information', *Computer Networks,* 54(12), pp. 120-122.

Cuevas, R. et al. (2010b) 'Book Section', in Kramer, J. D. (ed) *This is a another Book released in the same year,* London: PressAwesome, pp. 150-180.

Greene, D. & Williams, P. C. (1997) *'Linear Accelerators for Radiation Therapy',* 2nd edition, Bristol and Philadelphia: IOP Publishing Ltd.

Jacobson, J. & Andersen, O. (1997) 'Software Controlled Medical Devices', in *SP Report 1997:11,* Sweden: Swedish National Testing and Research Institute.

Ji, H. & Turner, K. J. (1999) 'Specification and Verification of Synchronous Hardware using LOTOS', in Wu, J., Chanson, S. T. & Gao, Q. (eds) *Formal Methods for Protocol Engineering and Distributed Systems (FORTE XII/PSTV XIX),* London: Kluwer Academic Publishers, pp. 295-312.

Turner, K. J. & Jennings, M. A. (2002) 'The Rules of Sailing Races for Hand-Held Devices', *J. Navigation,* 23(5), pp. 114-240.

University of Stirling (2002) *Computing Science and Mathematics Research,*   
http://www.cs.stir.ac.uk/research  
(accessed 01 March 2017).

The reference list above is generated by Word. Under ‘References’ -> ‘Manage Sources’ it is possible to add new references which can later be inserted in the report with ‘Insert Citation’. Example: (Cuevas, et al., 2010a)

Alternatively, you can add a new reference and immediately insert it as a citation by moving the cursor to the place where you want to insert the citation and then choosing ‘References’ -> ‘Insert Citation’ -> ‘Insert New Source’ from the ribbon menu.

You can hide parts of a citation by right-clicking on it, selecting ‘Edit Citation’ and ticking the checkboxes under ‘Suppress’. In the same dialog a reference to specific pages of the cited source can be added (which won’t be displayed in the references list). Example (same as above but with ‘Author’ suppressed and page 120 added): (2010a, p. 120)

Keep in mind that uncited sources will still appear in the references list above. Go to ‘References’ -> ‘Manage Sources’ to see which sources are cited and which aren’t. Sources in ‘Current List’ which have a checkmark are cited.

A custom style has been created for you to use to help with this. Download the file ‘HarvardUCLan2017.xsl’ and copy it to: %appdata%\Microsoft\Bibliography\Style.

Keep Word closed when doing this. To select the style click on ‘References’ -> ‘Style’ and select ‘Harvard – UCLan (2017)’. Tested with Word 2016 (Windows).

There might be some cases the Word bibliography function can’t handle. If you have a tool that suits you better such as ‘RefWorks’, ‘Citethisforme’ or ‘RefMe’ use that instead, then remove the list above and copy the references over. Don’t forget to use the correct notation style.

It is important that you refernece correctly, if you are not sure, ASK! You must be consistent, check your work!

Appendix 1 – Project Proposal

## Project Context

Multi-Room audio is typically an expensive platform to run; a premium product, with a price tag to match. SONOS, the market leader in multi-room audio setups, has several different networked speaker options – however I have some issues with them.

SONOS pricing starts at £179 for their basic speakers, and £399 for just the streamer unit if you want to use your own. Along with this, usage of the SONOS platform is limited to their app on iOS and Android, with a desktop application available for Windows. The system is *not* platform agnostic.

And while alternative solutions exist, such as Naim’s *Mu-so* system or Audio Pro’s C-series, all sport the same critical issues that plague SONOS speakers. More cost-effective alternatives include Amazon Echo or Google Home, however these also have drawbacks for my specific application. Both providers offer average audio quality with limited customization options for audio playback, but a larger issue for my application is the always-on voice monitoring. This may be a convenience for some, but due to privacy concerns that surround such systems it isn’t a desirable feature.

Even alternative “DIY” solutions such as Max2Play have their drawbacks. The only platforms that are capable of multi-room audio are limited by an entry fee, and there aren’t any viable open source alternatives. As an advocate for free and open-source software, this is something I want to change.

I believe it would be feasible to develop my own solution capable of streaming high-quality audio within a local network, without the previously stated drawbacks. The cost per-node can be as low as £25 (plus speakers), which is significantly less than the £399 speaker-less SONOS configuration.

JASLiN will be a platform-agnostic web-controlled LAN-based multi-room audio solution without the cost or gimmicks associated with either Sonos or Echo type systems. After the duration of this assessment, JASLiN will be released as an open-source project –free to anyone on the internet to modify, update, and upgrade.

## Specific Objectives

* To create a Raspberry Pi-based network audio streaming solution
* To develop a server system for controlling the audio stream
* To control the system from a platform-agnostic web-frontend
* To release the source-code openly for community development post-assessment

**References**

*Technical Documents and Papers*

Viewed 01 October 2019,<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.142.6102&rep=rep1&type=pdf>

AES, 2015. *AES standard for audio applications of networks - High-performance streaming audio-over-IP interoperability*, AES, viewed 01 October 2019,<http://www.aes.org/standards/comments/drafts/aes67-r-171107-draft-rev-cfc.pdf>

Cox, J., 2018. *Multi-room audio: everything you need to know | What Hi-fi?,* What Hi-fi?, viewed 01 October 2019. *<*<https://www.whathifi.com/advice/multi-room-audio-everything-you-need-to-know>>

Gstreamer, 2019. *GStreamer*, Gstreamer, viewed 01 October 2019,<<https://gstreamer.freedesktop.org/documentation/?gi-language=c>>

Humfrey, N. 2007. *MAST - Multicast Audio Streaming Toolkit*, Aelius, viewed 01 October 2019, *<*<https://www.aelius.com/njh/mast/>>

IANA, *Real-Time Transport Protocol (RTP) Parameters*, IANA, viewed 01 October 2019,<<https://www.iana.org/assignments/rtp-parameters/rtp-parameters.xhtml>>

Kuang, T., Williamson, C., 2002. *RealMedia Streaming Performance on an IEEE 802.11b Wireless LAN*, University of Calgary, viewed 01 October 2019,<http://pages.cpsc.ucalgary.ca/~carey/papers/2002/woc2002.pdf>

Locqueneux ,C., 2015. *How To Build A Multiroom Audio System Based On Raspberry Pi And Hifiberry*, HifiBerry, viewed 01 October 2019,<https://support.hifiberry.com/hc/en-us/articles/205699981-How-to-build-a-multiroom-audio-system-based-on-Raspberry-Pi-and-Hifiberry>

Majumba, A., et al., 2002. *Multicast and unicast real-time video streaming over wireless LANs*, viewed 01 October 2019,<https://ieeexplore.ieee.org/abstract/document/1013857>

Smith, N. & Farkas, L., 2019. GitHub - gstreamer-java/gst1-java-core: Java bindings for GStreamer 1.x, Codelerity Ltd, viewed 01 October 2019, *<*<https://github.com/gstreamer-java/gst1-java-core>>

Sonos, 2019, *Sonos,* Sonos, viewed 01 October 2019,<<https://www.sonos.com/en-gb/home>>

## Potential Ethical or Legal Issues

Legal issues may primarily concern licensing agreements with third-party libraries (EG: GStreamer). If this product were to go to market, this would be of even greater consideration, as there may be potential issues if the deliverables were to include the required hardware to operate. If system logs are to be generated by the server or client nodes as a result of the program, the data collected would need to be handled carefully and anonymised to ensure that data regulations within the EU (and the customer’s country) are adhered to. This applies to customer details too, assuming any are stored. Many of these issues can be mitigated by ensuring the licenses of any integrated libraries are adhered to – as JASLiN is planned to be freeware, many of the concerns regarding resale of software packages are alleviated.

Ethically speaking, there are several similar products that operate within a similar domain, for example Sonos systems, and this may pose an issue by means of cutting into their market share. This can be mitigated by not using or implementing any resources published by Sonos or their competitors, and by ensuring any designs, features, or concepts are not lifted from their work. The modern free market allows and encourages competition by design.

The system and source code, once finished, will be published under a presently undecided open-source license which should pose no ethical issues, however may present legal ones depending on the libraries and resources implemented within the project.

## Resources

*Documentation Resources*

GStreamer - <https://gstreamer.freedesktop.org/documentation/tutorials/index.html?gi-language=c>

GStreamer Java - <https://github.com/gstreamer-java/gst1-java-core>

Debian Java - <https://wiki.debian.org/Java/>

Multicast Audio Streaming Toolkit - <https://github.com/njh/mast>

MAST - <https://erg.abdn.ac.uk/users/gorry/course/intro-pages/uni-b-mcast.html>

FFMPEG - <https://www.ffmpeg.org/documentation.html>

Java UDP - <https://docs.oracle.com/javase/tutorial/networking/datagrams/index.html>

Python UDP - <https://wiki.python.org/moin/UdpCommunication>

NodeJS - <https://nodejs.org/en/docs/>

React - <https://reactjs.org/docs/getting-started.html>

Material UI - <https://material-ui.com/>

AnimeJS - <https://animejs.com/documentation/>

*Online Libraries/Lectures*

LinkedIn Learning, formerly Lynda.com - <https://www.linkedin.com/learning/>

Udemy - <https://www.udemy.com/>

Skillshare - <https://skillshare.com/>

*Tutorial Resources*

Running C on Pi - <https://www.raspberrypi.org/magpi/learn-code-c/>

Running Java on Pi - <https://javatutorial.net/raspberry-pi-java-tutorial>

Developing IoT applications for Pi - <https://dzone.com/refcardz/iot-applications-with-java-and-raspberry-pi?chapter=1>

Java Media Framework - <https://www.oracle.com/technetwork/java/javase/tech/index-jsp-140239.html>

Java UDP Tutorial - <https://www.codejava.net/java-se/networking/java-udp-client-server-program-example>

ReactJS Tutorial - <https://reactjs.org/tutorial/tutorial.html>

pHAT DAC Tutorial - <https://learn.pimoroni.com/tutorial/phat/raspberry-pi-phat-dac-install>

*Software Resources*

Jetbrains IntelliJ IDEA - <https://www.jetbrains.com/idea/download/#section=windows>

Visual Studio Code - <https://code.visualstudio.com/>

Draw.io - [http://draw.io](http://draw.io/)

GIMP - <https://www.gimp.org/>

Inkscape - <https://inkscape.org/>

Git Fork - <https://git-fork.com/>

Github - <https://github.com/>

*Other Resources*

Red5 Media Server - <http://red5.org/>

Icecast - <http://icecast.org/>

Ampache - <https://github.com/ampache/ampache>

## Potential Commercial Considerations - Estimated costs and benefits

There are several different factors for consideration regarding the potential commercial and financial viability of the project. Below is a tabular compilation of the hardware and software requirements for the project endeavour; as the project is intended to be freeware, the cost of demonstration hardware will be provided by myself – however, if I were to charge a customer for a bespoke project such as this, they would provide both the cost of the hardware and the cost of my time, as approximated below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Item** | **Purpose** | **Cost** | **Quantity** | **Measure** | **Total** |
| **Hardware** | | | | | |
| Raspberry Pi 4 – 2GB | To act as the Server for the audio stream and web control interface | £45 | 1 | Unit | £45 |
| Raspberry Pi Zero W | To act as Client node(s) for receiving the audio stream | £15 | 1 | Unit | £15 |
| PHat DAC | To output audio from the client nodes | £15 | 1 | Unit | £15 |
| 5v Power Supplies | To provide power to Server and Client node(s) | £5 | 2 | Unit | £10 |
| Cables (assorted) | For connecting devices together | £3 | 5 | Unit | £15 |
| 8GB Micro SD Cards | To store the Operating Systems for the Server and Client node(s) | £5 | 2 | Unit | £10 |
|  |  |  |  |  |  |
| *Hardware Cost* |  | | | |  |
| **Software** | | | | | |
| IntelliJ IDEA Community | To develop Java applications | £0 | 2 | Installation | £0 |
| Visual Studio Code | To develop web applications | £0 | 2 | Installation | £0 |
| Draw.io | To design charts and diagrams | £0 | 2 | Installation | £0 |
| GIMP | To create raster graphic assets | £0 | 2 | Installation | £0 |
| Inkscape | To create vector graphic assets | £0 | 2 | Installation | £0 |
| Git Fork | To manage git repositories | £0 | 2 | Installation | £0 |
| *Software Cost* |  | | | | *£0* |
| **Services** | | | | | |
| Development | Time invested in developing the software | £15 | 400 | Hours | £6,000 |
|  |  |  |  |  |  |
| *Service Cost* |  | | | | *£6,000* |
| Total Cost |  | | | |  |

While the software has potential marketability, I want to undertake this project to release an open-source alternative. This means that there will be no expected return on investment for the alproject, unless a benefactor voluntarily donates. By doing this, it will benefit the software community by offering an alternative to the current paid-for platforms, and those who want an inherently extensible audio streaming platform. The deliverable artefacts produced throughout this project are not inherently time-bound, thus there is no incentive to rush the development process.

## Proposed Approach

The project will be tackled in five primary stages in order to achieve the core functionality of the software. The lifecycle methodology I will use to implement this will be Agile-based.

Core Functionality is the Minimum Viable Product (MVP) for the project. This will be defined as the following features: Server node capable of broadcasting audio over the network; Multiple client nodes capable of simultaneous network audio playback; Server node capable of controlling the broadcast audio (Change track, play/pause, etc.); User capable of controlling the broadcast audio via a web frontend

20 hours will be invested into designing the solution architecture. 60 hours will be allotted for casting a constant looping audio stream from the server, and another 60 will be allotted for receiving the stream on the client nodes. Afterwards, another 60 hours of work will be put into creating the server backend to control the audio stream, and a further 50 hours for developing the web UI and frontend control mechanisms for the user. After this, I expect testing and system integration to take approximately 50 hours, for a cumulative estimate of 280 hours for the core functionality. This leaves a bucket of approximately 100 hours as either overhead in case of unexpected difficulties in implementation, or to be used for developing and implementing additional functionality to the system.

As of writing this, there are 31 weeks until May 2020 and the System will require approximately 9 hours of work per week to achieve core functionality, not accounting for additional time from the overhead.

Additional Functionalities are those that are outside the scope of the MVP, and will be later analysed and categorised by their desirability and ETA for implementation within the system. In the immediate term, it will be assumed that each additional feature will require between 15 and 20 hours of work to implement successfully. Examples of such additional features include the following:

Room grouping; Channel separation; Per-node volume; User authentication; Playlist management; Display for current playing track; Play/pause/skip (hardware based); Upload songs via web interface; Upload songs via SSH; Upload songs via USB; Cache library of songs; Spotify integration; Calculate delay playback; DSP/EQ; Bluetooth input; and Stereo/Surround capabilities.

Appendix 2 – Technical Plan

Your second appendix should be a copy of your Technical Plan.

You may have one or more appendices containing detail, bulky or reference material that is relevant though supplementary to the main text: perhaps additional specifications, tables or diagrams that would distract the reader if placed in the main part of the dissertation. Make sure that you place appropriate cross-references in the main text to direct the reader to the relevant appendices.

Do not blindly include all of your code in the appendix or the body. Only include the parts you refer to in the report. You can put those parts either in the appendix or in the body (e.g. in the “Implementation” part).

Appendix 3 – Title of Appendix

You may have one or more appendices containing detail, bulky or reference material that is relevant though supplementary to the main text: perhaps additional specifications, tables or diagrams that would distract the reader if placed in the main part of the report. Make sure that you place appropriate cross-references in the main text to direct the reader to the relevant appendices.

Do not blindly include all of your code in the appendix or the body. Only include the parts you refer to in the report. You can put those parts either in the appendix or in the body (e.g. in the “Implementation” part).