**Music-Synced Smart Lighting System**

A dissertation submitted in

partial fulfilment of  
The requirement for the degree of  
**MASTER OF SCIENCE**

in

**Software Development**

in  
**The Queen’s University of Belfast**  
By

**Ryan Urquhart**

**40099112  
06/05/2025**

**Acknowledgements:**

* Bee-Yen Toh – My supervisor who was with me every step of the way
* Cherise McClean – My wife, a constant (and welcome) distraction
* SoDes – On GitHub, created a package that made this whole project possible on windows (PyAudioWPatch)
* aiXander – On GitHub, created a repo that I adapted, it provided fast Fourier transformations that I built the beat detection algorithm on top of.

Abstract:

This report covers the development of a Music-Synced Smart Light System. The development of this system was successful, however, due to time constraints the system is missing many aspects that would be desirable should the project be continued on. This is aided by the style of development of the system, the system has been developed in such a manner as to allow easy continued development. The system takes advantage of FFT of audio data from PyAudio to allow the system to detect the beat of music and change bulb brightness in time with that beat.

Table of Contents

[Introduction: 5](#_Toc195705077)

[1. Understanding the Problem 6](#_Toc195705078)

## Introduction:

This report is focusing on the development of a Music-Synced Light Control System (hereafter referred to as, Bulb Bop). The report will consist of six chapters, Chapter One, Understanding the Problem, Chapter Two, User interface-design, Chapter Three, Architecture design and algorithm explanation, Chapter Four, Experimentation, Chapter Five, Testing, Chapter Six, Evaluation and Conclusion. The content of these chapters will be explained in the rest of the introduction.

Chapter One will contain my exploration of the problem and the basis on which I started undertaking the development of the software system. This will include my initial understanding of the project, based on the title and the preliminary meetings with my supervisor. It will then progress into my initial investigations about the potential solutions to the problem at hand. Finally, this chapter will end with the decided upon plan of action for development.

Chapter Two will contain the plan and the development of the user-interface. This will include the early iterations of the user-interface, and how these early iterations influenced the final design. It should be noted however, that the bulk of the work in this project was the beat detection algorithm and therefore, there was less thought put into the user-interface than there was into the back end of the system.

Chapter Three will contain the plan and the development of the architecture and algorithm design. This section will explain the architecture upon which the system can be ran, and the algorithms that allow it to run. This is the chapter in which the bulk of the work of this project will be explained.

Chapter Four will contain the experimentation that I have conducted due to this project. This will include looking at different languages that I attempted to utilise, as well as the research and experimentation that went into the development of the beat detection algorithm. Various iterations and experiments will be present in this chapter.

Chapter Five will contain the testing carried out on the project, during its development, and before the submission of the project. This will include risk assessments of the project, and a discussion of the rationale of the tests that were designed, as well as an examination of the areas of the project that were not thoroughly tested and an explanation as to why.

Chapter Six will be the final evaluation and conclusion of the project. Discussing the success or failure of the development of the system. It will evaluate different technologies utilised and how they have impacted the final system shape. There will be a reflection on my own development of the project, highlighting areas that were handled well, and areas where there could have been further development.

# Understanding the Problem

The initial description of the project is as follows,

“Develop a system that integrates music with dynamic lighting displays to create an "instant smart disco light" function or a “smart lighting solution for improved well-being.” The system should control a Wi-Fi-enabled smart light bulb, adjusting its brightness and colour in sync with music rhythms, enhancing the atmosphere in any setting.”

As well as this, there were four main functional criteria for the development of the system.

1. The system should connect to and control a Wi-Fi enabled smart light bulb
2. The system should detect music rhythm and tempo from an audio source and synchronise the light settings accordingly
3. The system should provide a primary control interface
4. The system should offer multiple lighting modes

There was no restriction placed on the technologies to be utilised in this project. No specific bulbs, programming languages, or web frameworks. However, due to having to use third-party hardware (The smart bulbs) there may be a specific API dictated by the manufacturer that would have to be used.

The initial prompt has no definition of the user the application is supposed to be designed for. However, during meetings with my supervisor the intended user was defined in a varied way. From families, to DJ’s, to people hosting a party. The decided goal was that the user would be someone hosting a party in their house, this user would be either using a microphone attached to the computer running the system, or the computer running the system would be playing the music from its speakers.

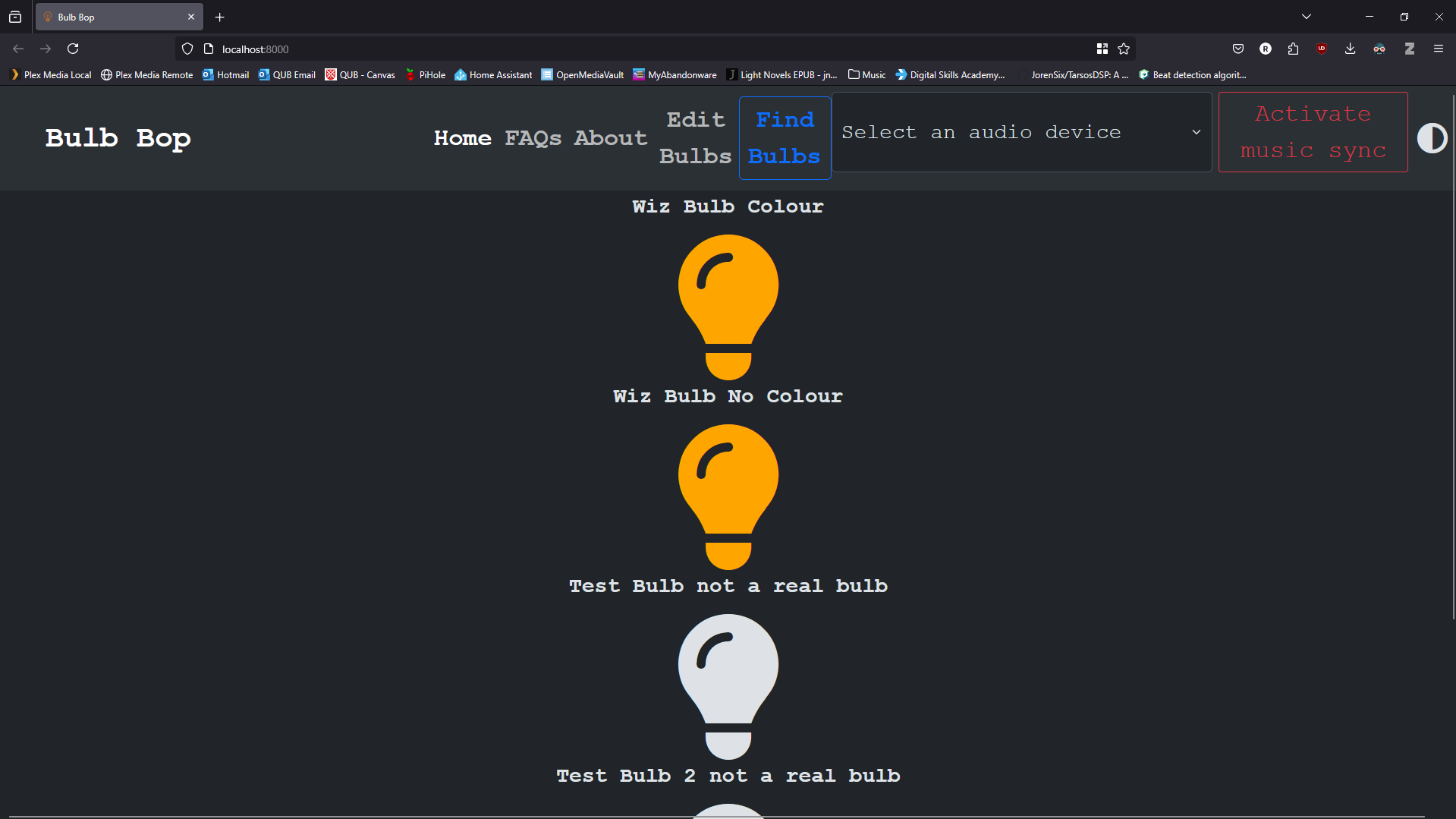
From the initial description of the project, research was carried out, this research covered two main areas. The first was the bulb, which brand would be appropriate for the project. The second was the programming language, this programming language had to fulfil two criteria, it needed access to the audio from the operating system I was developing on (Windows and Linux), it also needed to be able to host a web server which would provide the user interface.

The research into the bulbs brought up many different options for use, Wiz, TP-Link Kasa, TP-Link Tapo, YeeLight, Shelly, Philips Hue, Athom.tech, and switchbot. From all of these bulbs I decided to use Wiz bulbs. The reasons for this were the cost, the open API (*bulbs like TP-Link Tapo have closed API’s so users must use their app*), the app does not require a hub to operate (*Philips Hue does*), and from my reading of various forums wiz was highly recommended as a bulb brand. Based on this research I purchased a WizBulb A60.E27 bulb.

The research into the programming language to use initially started with Java, the reason for this simply being that much of the CSC7063 Computer Programming course has been taught in Java. Two libraries I looked at were TarsosDSP(Six, 2011/2025), and Jipes by Tagtraum(Schreiber, 2013/2024), both of which I was unable to gain access to audio data from the host machine using. Jipes was unsuited as it was not made for real-time processing of audio information. With regards to TarsosDSP, it used Gradle to build it, this was something that, given the time I needed to get the project started in, I was unable to figure out in time.

As such, I looked for another programming language that would allow me access to this host audio data. For this I turned to python, there is a python package names, PyAudio, and it reliably and easily allows developers access to the host machine audio data. The issue however, is that PyAudio does not allow access to the loopback api present on windows machines that allows the developer access to real-time speaker audio data. This was solved when I found a windows specific patch of the PyAudio library, called PyAudioWPatch. This allows PyAudio the ability to access the loopback api, this with the Django library that is able to serve web apps with a great deal of flexibility made python the perfect choice.

# User Interface Design

A screen shot of a computer

AI-generated content may be incorrect.

Figure 1 - Computer view of index page Figure 2 - Phone view of index page

The user interface for this system is quite simple. There are a total of four different views that the user can access. Each of these views are dynamic to varying degrees, Django gives the option to develop html that will dynamically change based on a context provided by the developer. For instance, the bulbs on the index page render depending on if they are present in the database or not.

The first part of the user interface that we will discuss are the parts common to all views. The main part of this is the navbar, this contains the title, links to other views, buttons to activate functions of the system, and a light/dark mode toggle button.

**SCHOOL OF ELECTRONICS, ELECTRICAL ENGINEERING and COMPUTER SCIENCE**

**CSC7058 – INDIVIDUAL SOFTWARE DEVELOPMENT PROJECT**

A signed and completed cover sheet must accompany the submission of the Individual Software Development dissertation submitted for assessment.

Work submitted without a cover sheet will **NOT** be marked.

**Declaration of Academic Integrity**

Before signing the declaration below please check that the submission:

1. Has a full bibliography attached laid out according to the guidelines specified in the Student Project Handbook
2. Contains full acknowledgement of all secondary sources used (paper-based and electronic)
3. Does not exceed the specified page limit
4. Is clearly presented and proof-read
5. Is submitted on, or before, the specified or agreed due date. Late submissions will only be accepted in exceptional circumstances or where a deferment has been granted in advance.
6. Software and files are submitted via Canvas.

**I certify that that the submission is my own work, all sources are correctly attributed, and the contribution of any AI technologies is fully acknowledged.**

**I declare that I have read both the University and the School of Electronics, Electrical Engineering and Computer Science guidelines on plagiarism - http://www.qub.ac.uk/schools/eeecs/Education/StudentStudyInformation/Plagiarism/ - and that the attached submission is my own original work. No part of it has been submitted for any other assignment and I have acknowledged in my notes and bibliography all written and electronic sources used.**

**I am aware of the disciplinary consequences of failing to abide and follow the School and Queen’s University Regulations on Plagiarism.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name: (BLOCK CAPITALS)** |  | | |
| **Student Number:** |  | | |
|  |  |  |  |
| *Student’s signature* |  | *Date of submission* |  |