ÇANKAYA UNIVERSITY FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

CENG 407

PROJECT REPORT

AMBILIGHT MEDIA PLAYER

Utku YILMAZ 201911406

Ayhan ARICI 201811252

Elif AKÇAYIR 201811251

Emre Can AŞIK 201811291

Table of Contents

1. Literature Review

1.1.1. Öz

Günümüz dünyası her geçen gün daha da modernleşmektedir. Akıllı cihazların hayatımızdaki konumları her geçen gün genişlemektedir. İletişim, temizlik, sağlık, eğlence, finans gibi birçok alanda akıllı cihazlardan faydalanmaktayız. Bu alanlarda ilerlemeler gerçekleştikçe artık evlerimizde de bu ihtiyacı hissetmeye başlamamız kaçınılmaz olmuştur. İnsanlar artık kendileri evde değilken evin temizlenmesini, havasının filtrelenmesini, sahip olduğu ürünlerin tazeliğini kontrol edebilmeyi kendilerince gereksinim olarak görmektedir. İnsanlar sadece bu teknolojiye sahip olmayı değil göze de hitap etmesini istemektedir. Günümüzde çoğunluğu gençler olmak üzere teknolojiye bağlı kalmayı başarabilmiş insanların evlerindeki bilgisayar, kulaklık, aydınlatma sistemleri, televizyon gibi birçok ürünün göze güzel gelmesi önemli bir kriter oluşturmaktadır. Bu ihtiyaçtan yola çıkarak bahsedilen ürünlerin birçoğunda RGB teknolojisi kullanılmaktadır. Bu proje insanların günlük hayatta kullandıkları bilgisayarlarından videolar izlerken ortamın bu videolardaki renkler kullanılarak aydınlatmasını ve bunu dinamik bir şekilde sağlamasını amaçlamaktadır. Kullanıcı ortam aydınlatması sayesinde kendini izlediği filmin, dizinin veya videonun içerisinde hissedebilir. Ortam aydınlatmasını masaüstü uygulaması aracılığı ile kontrol edebilecek ve kendi tercihlerini uygulayabileceği seçimler yapabilecektir.
Bu literatür taraması ile duyulan gereksinime yönelik bir sistemin nasıl tasarlanacağını, karşımıza çıkacak sorunlara çözümler üretebilmeyi, kullanışlı ve fiyat/performans dengesine sahip bir ürün ortaya çıkarabilmek için katmamız gerekenleri öğrenmeyi ve önceki çalışmaların özetini sunmayı hedefliyoruz.

Anahtar Kelimeler: Ortam Aydınlatma, RGB, Masaüstü Uygulama

1.1.2. Abstract

Today's world is becoming more and more modern every day. The position of smart devices in our lives is constantly expanding. Smart devices are used in many areas such as communication, cleaning, health, entertainment and finance. As progress has been made in these areas, it has become inevitable that this need will begin to be felt in homes. People no longer consider it necessary to clean the house, filter the air, and control the freshness of the products they own when they are not at home. People want not only to have this technology, but also to appeal to the eye. Today, it is an important criterion that many products such as computers, headphones, lighting systems, televisions in the homes of people who have managed to stay connected to technology, mostly young people, are attractive to the eye. Based on this need, RGB technology is used in many of the products mentioned. This project aims to illuminate the environment by using the colors in these videos while watching videos from the computers that people use in their daily life and to provide this dynamically. Thanks to the ambient lighting, the user can feel himself in the movie, series or video being watched. It will be possible to control the ambient lighting through the desktop application and make choices where preferences can be applied.
With this literature review, we aim to learn how to design a system for the needs, to find solutions to the problems we will encounter, to learn what we need to add in order to come up with a product that is useful and has a price / performance balance, and to present a summary of previous studies.

Keywords: Ambient Lighting, RGB, Desktop Application

1.1.3. Introduction

Today, technological products are designed to meet many needs of people. With the emergence of these technological products, desktop or mobile applications have also been developed to control them. Our goal is to enable users to easily reach the environment they want, thanks to the video ambient illuminator. With this application, it can be selected whether the lamps will work or whether they will be colored in sync with the video. In this way, you can have a more realistic and eye-catching experience. Moreover, the screen part of the lamps can be selected via the application and changes can be made via IP addresses. The program running in the background will instantly capture the images on the screen, find the color tones and send instructions to the necessary lamps to reflect these colors. Thanks to this application, you can bring the cinema to your home and eliminate screen borders. You can set your mood for that day by using this technology in your living room, living room or bedroom. For example, you may want to feel yourself on the green field while watching a football match, or you may get lost in the blue while watching the ocean. Some television brands may have similar technology, but computer monitors do not have this feature. It does not make sense to use those technologies in computer monitors in terms of price. While this system works in an embedded way in television products, it will work with open source code in our project. From this point of view, it stands out compared to others in terms of cost and usage.

We will be using the Python Programming Language for our project because this way we can create a Linux/Windows based project. In addition, Yeelight, the lamp we want to use, has its own Python library. The fact that it is more stable and readable compared to other languages has also been effective in this choice. Our team has experience with Python and is in constant communication with each other. Ultimately, we chose this language because the software environment we intend to develop and intermediary programs such as VLC, whose libraries we will use to play the video, interact well with Python.

1.2. Development Environment

1.2.1. Python

Python [1] is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems. This versatility, along with its beginner-friendliness, has made it one of the most-used programming languages today. [2] Python on Windows, macOS, and Linux operating systems. Python can help with build management, bug tracking, and testing. Software developers can use Python to automate testing for new products or features. This was one of the reasons of our choice of Python as envoriment.

1.2.2. Yeelight API / Library

The YeeLight Python library is a small library that lets you control your YeeLight RGB LED bulbs over WiFi. [3] Yeelight smart devices allow users who are willing to play with the device with their own softwares.

1.2.3. K-Means

Clustering is a collection of methods for dividing data into groups, or clusters. Clusters are roughly described as collections of data objects that are more comparable to each other than to data objects from other clusters. [4] Kmeans Algorithm is an Iterative algorithm that divides a group of n datasets into k subgroups /clusters based on the similarity and their mean distance from the centroid of that particular subgroup/ formed. In our project, we will use this python library to find the color intensities in the image we get from the video, and to convert them into a result by clustering. [5]

1.2.4. PyQt5

PyQt is a GUI module that integrates the Qt C++ cross-platform framework to the Python language. Qt is more than just a graphical user interface toolkit, which is why it includes abstractions for network sockets and threads, as well as Unicode, SQL, databases, SVG, OpenGL, XML, an operational web browser, a service system, and a large number of GUI widgets. [6] Also Qt Designer tool can be used to create and develop an interface for Ambilight Media Player.

1.2.5. OpenCV

OpenCV is a large open-source library for image processing, computer vision, and machine learning. Python, C++, Java, and other programming languages are supported by OpenCV. It can analyze images and videos to recognize objects, faces, and even human handwriting. [7] Many organizations, including Google, Amazon, Microsoft, and Toyota, use this library today. A large number of scholars and developers participate. It is simple to install on any operating system, including Windows, Ubuntu, and MacOS.

1.2.6. Flutter

Flutter is a Google-developed open source user interface platform. It is used to create cross-platform applications from a single codebase for Android, iOS, Linux, Mac, Windows, Google Web Platform, and the web. The Flutter framework is made up of two parts: a software development kit and a widget-based user interface library. Sliders, buttons, and text inputs are among the reusable UI elements in this package. Developers working with the Flutter framework will use the Dart programming language to create mobile apps. [8] Dart is a typed object programming language that focuses on front-end development, with a syntax similar to JavaScript.

1.3. Related Works

1.3.1. Philips Ambilight

Rows of LED bulbs are integrated into the edges of a Philips Ambilight TV. Ambilight 2, 3, and 4 light up 2, 3, or 4 sides of the screen, correspondingly. The TV's lighting can be set up in a variety of ways. The LED lamps light up in the colors of the screens when the screen mode is selected. In music mode, the lighting changes to match the beat of the music.[9] The company applies this technology only to televisions. You have no choice but to buy the product. Thanks to our application, we will make a similar technology available on TV screens or monitors.

1.3.2. Lightberry

Lightberry is a plug-and-play RGB LED kit for Raspberry Pi that is simple to use. Everything you'll need (except the Rpi) is included in the box, and it's ready to use without the need for soldering. Lightberry can be used in a variety of ways, but it's most commonly placed behind the TV to create ambient light that adjusts to what's on screen.[10] Lightberry is compatible with any Raspberry Pi video (i.e. MKV files, video streams like youtube).
Lightberry can also be utilized with an external HDMI source (blu-ray player, game console, etc.) however this arrangement requires the purchase of a separate component – an HDMI Converter. The downside of this technology is that you need to have a Raspberry Pi to use it. This is the problem in terms of the overall cost of the product. Even if you have the necessary products, there are stages such as software setup and plugin configuration, and it is observed that it is not a user-friendly system for people who are not very interested in technology. While we are creating our project, we aim to correct these aspects as well.

1.3.3. Dreamscreen Backlight

DreamScreen is a Smart TV Lighting Kit with LEDs that react to what you're seeing on your TV. DreamScreen enlarges your TV, softens the picture, and makes your favorite TV, Video Games, and Movies bigger and better. It adjusts to any size TV and works with any HDMI source. WS2812B, SK6812, and WS2813 digital RGB LED lights are presently supported by DreamScreen DIY.[11] The total number of LEDs is limited to 250. Each side may contain 45 people, with the top and bottom each having 80. Because this is a DoItYourself(DIY) product, the user must purchase the leds separately and perform the installation themselves. Although it is not

widely used due to size and LED constraints, it has reached a point where it can be used as a temporary substitute. The high standards and labor redundancy stand out as drawbacks. Since the product is also presented as a kickstarter project, the support team and sales team are weak compared to other product providers.

2. Software Requirement Specification (SRS)

2.1. Introduction

Ambilight is defined as the fluorescent technology on the rear edge of the television that can light up according to the colors on the screen. Ambilight televisions reflect the color of whatever is on the screen while watching a movie or playing a game on the TV in the evening. In Ambilight technology, the LED lights on the back of the television work in harmony with the screen. Whatever color is on the edges of the screen, the lights in that area are lit in the same color. Thus, the ambiance of the image user watch on television is strengthened and it is aimed to increase the pleasure user receive. These processes are realized thanks to the embedded system in televisions with Ambilight technology. It works simultaneously by reacting to color weights and transitions in certain areas of the screen. It works with the algorithm of capturing the image, processing and reflecting the color. The importance of coloring and visual beauty in today's living spaces is increasing day by day. We believe that its use will increase in the near future due to the need to keep up with the times and follow innovations closely. Our project will be designed with the aim of presenting the same process on monitors in a more accessible and less costly manner. In this part of the report, we explained our purpose, scope of our project, target audience characteristics, glossary and finally an overview of our SRS document.

2.2. Purpose of This Document

In this project, our aim is to produce an alternative media player that works with Ambilight logic by using smart ambient lighting bulbs. This media player will transmit the color illuminations of the current screen to the bulbs and create an environment. It will detect the bulbs in the network via the software and make them ready for use for the user. After the media file to be played is specified to the program, the video will start playing and the bulbs will react instantly. In this way, we plan to create a product that is much more reasonable in terms of price compared to the technology used in televisions and can compete in terms of performance. While doing the project, we aim to design Ambilight Media Player using various software requirements such as Object Oriented Programming, Internetworking, Socket Programming. In the continuation of the project, we aim to test our system as a whole by using the smart ambient lighting bulbs, which are the hardware part of Ambilight Media Player, simultaneously. We see it as our goal to detect the intensity of the colors in the media files according to the plots through the Ambilight Media Player software we will create, to ensure that the obtained color values are transmitted to the bulbs determined through the system simultaneously, and to ensure that these processes continue in an uninterrupted cycle until the video is completed. While the system is running, it will take the codes of the colors that appear on the screen simultaneously with the media file and send it to the system. Thanks to the libraries and developer kits we will use, the right value of light will be reflected from the bulb at the right spot. This SRS document contains the project requirements and software requirements specification for Ambilight Media Player.

2.3. Scope of the Project

This project aims to illuminate the environment by using the colors in these videos while watching videos from the computers that people use in their daily life and to provide this dynamically. Our project will not only provide ambient lighting according to the video provided, but also support the control of the bulbs, which are located at four different angles, through the interface, through the software we will develop. It will be possible to switch between the videos in the source via the software and make sound adjustments. It will be worked on that the bulbs on the network are automatically detected by the system and assigned to the bulbs. The resulting product will be designed to appeal to all types of users. Thanks to its user-friendly interface, the user will be able to access all functions without the need to use an extra computer. After the bulbs are placed in the sockets and defined in the network, the user will be able to have all the features found in Ambilight televisions by simply using the software. One of the aims of the project is to establish a system that is easily understandable to install and use. We aim to establish a mechanism where the user can perform tests and give feedback when necessary. In the first part of our project, we will concentrate on the software and perform the necessary programming processes to determine the colors on the screen, their intensity, the bulb to be transmitted according to their location, and to record and send them to the bulbs simultaneously. In the next part of our project, we will include the hardware part and enable it to interact with the software we have prepared, and we will have simultaneous and multiple operations.

2.4. Glossary (Definitions, Acronyms, and Abbreviations)

TERM	DEFINITION
Actor	An actor can be a user, or hardware that interacts with the system.
Windows	Windows is an operating system designed by Microsoft. The operating system is what allows user to use a computer.
Linux	Linux is a free, open-source operating system, released under the GNU General Public License (GPL).
Python	Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis.
Yeelight / API	Yeelight Smart LED Bulb is an affordable color LED bulb that user can control. Yeelight Python library is a small library that lets user control bulbs over Wi-Fi.
K-Means	Kmeans is an Iterative algorithm that divides a group datasets into subgroups based on the similarity and their mean distance from the centroid of that particular subgroup.
PyQt5	PyQt is a GUI module that integrates the Qt C++ cross-platform framework to the Python language.[1]
OpenCV	OpenCV is a large open-source library for image processing, computer vision, and machine learning.[2]

Software Requirement Specification (SRS)	A document that provides comprehensive description of system's functions, requirements, constraints, and conditions to be able to perform. This document is an SRS document.

2.5. Overview of Document

This document contains three main topics.

- The first topic gives information about the document, such as purpose of the document and the terms used in the article.
- The second topic introduces the system. Features of the functions used, user classes and requirements for creating the Ambilight Media Player system.
- The final title gives a detailed introduction to the requirements and explanations of the requirements in detail.

2.6. Overall Description

This part will clarify of the principal aspect of Ambilight Media Player system and necessities.

2.7. Product Perspective

The product is supposed to be an open source, under the GNU general Public License. There are two separate sections in our system as hardware and software. In our project, we will use computer science concepts such as image processing, computer vision, and machine learning. While using them, we will make use of the Python programming language. We will use Python IDLE

throughout the development process. In addition, we will also benefit from several different development tools. In this section, we have summarized which features are included in terms of hardware and software. In our project, there are 4 Xiaomi Yeelight 1s YLDP13YL smart led bulbs, 4 E27 bulb holders, 1 monitor that supports 1920x1080 resolution that provides full HD quality screen display. We use Python IDLE and Pycharm as software development IDE. We use Yeelight's API[3] library to run our code on the network. We aim to benefit from the functionality of libraries such as OpenCV, PIL (Python Imaging Library), PyQT5, Yeelight, K - Means in order to detect bulbs and to realize continuous data flow. The information of the bulbs can be entered into the program manually or it can be requested to be detected and processed automatically by the program. A 2.4Ghz internet connection is required for the transfers in the network to be stable and supported by all tools in the system.

2.8. Product Functions

Open Video: There is a media player embedded in the system. When the button is activated, it directs the user to select a media file from the file directory. After the video is selected, the program is ready to play the video on its own interface.

Play Video: It allows to play the video that is ready on the program. The video can be played regardless of whether the bulbs are defined on the program.

Pause Video: It can pause the running video momentarily. If the bulbs are connected, the lighting will continue to reflect the colors of the scene that is stopped momentarily.

Stop Video: Allows the video being played to be completely stopped and rewinded to the beginning.

Go to the first scene: Allows user to return to the first scene of the video that is running on the program.

Go to the next scene: Allows user to go to the next scene of the video that the program is working on.

Go to the previous scene: Allows the program to return to the previous scene of the video

currently working on it.

Go to the last scene: Allows user to go to the last scene of the video that the program is working on.

Full Screen: Allows the video being played to fill the entire screen by formatting itself according to the screen resolutions of the desktop.

Mute/On Sound: Allows user to completely mute or unmute the currently playing video.

Volume Setting: Allows user to increase or decrease the volume of the video being played.

Activate Bulbs: It must be activated in order for the defined bulbs to reflect color simultaneously with the video.

2.9. User Classes and Characteristics

In this section, actors of our project are listed. There are 3 different actors in our project.

2.9.3. User

User is the actor who is responsible for starting the program and getting the bulbs ready for use. User's tasks are as follows:

- Manually starts the Ambilight Media Player software.
- Selects the media file to be played.
- Makes bulbs ready for use by making them available on the network via Ambilight Media Player.
- Checks whether the bulbs are working.
- Runs the program.

2.9.4. Ambilight Media Player

Ambilight Media Player is the actor that establishes the relationship between the user and the lighting system and ensures that the media file is processed and sent to the bulbs. AMP's tasks are as follows:

- Makes the video playable on the interface after the media file is selected.
- Performs the operations to play, pause, pause or rewind the video.
- Finds and distinguishes the bulbs on the network and records their IP addresses in the system.
- By sending a pre-prepared fixed code to the bulbs, makes them light up in a certain cycle and reports the result of the test.
- While the video is playing, it records the color values on the screen in the background and processes these values and transmits them to the bulbs correctly.

2.9.5. Smart Bulb

Smart bulb is the actor that receives the color codes obtained from the video by AMP over the network and reflects the color of that code[4]. Smart Bulb's tasks are as follows:

- Stays connected to the Network over Wi-Fi.
- Simultaneously processes the color codes sent by the AMP and reflects it as light.

2.9.6. Screen

Screen is the actor on which the program runs and the video plays.

Screen's tasks are as follows:

- Allows display of operating system and program.
- It performs the physical communication between the user and the software.

2.10. Operating Environment

In order to perform the relevant functions, the characteristics of the system should be as follows:

- The operating system on which the program is installed must be Windows or Linux. The operating system must be connected to electricity.
- The operating system on which the program will run must be connected to the Internet.
- Bulbs must be connected to the internet.
- Bulbs and AMP must be on the same network.
- Bulbs must be on.

2.11. Constraints

This system is built on the extremely up-to-date PyQt5[6]. Which bulb to use is decided by specifying that data exchange or data transmission will provide efficient performance. There are some restrictions for the system to work properly.

- If no bulbs can be detected on the AMP, it will report this to the user as a warning.
- If the selected media file is not suitable for playback, the user is warned.
- Since our software is open source code, it is only allowed to be installed from a trusted source.
- Bulbs and operating system must be connected to the internet via 2.4Ghz Wi-Fi.

2.12. Dependencies and Assumptions

In this part, we have listed all of our system's dependencies and assumptions, along with brief explanations.

Environment Conditions: The system will work properly while the operating system and hardware parts are smooth. The system may not work properly when there are weak internet connection, CPU shortage, unstable voltage etc. It is also assumed that the operating system is Linux.

Operating System: The system will be supported by Windows and Linux. System must have 2.4GHz communication frequency via Wi-Fi. Python, Yeelight, K-Means, PyQt5, OpenCV libraries must be installed on the system via PIP Install Packages or via executable file, as stated earlier. Mobile application needs Android 4.4 iOS 8.0 for the products setup.

Hardware: There are several hardware required for AMP to work properly. Our program is designed to work optimized with Xiaomi Yeelight 1s YLDP13YL. Since the working principles are the same, it is also compatible with Xiaomi Yeelight M2-E27 YLDP26YL and Xiaomi Yeelight W3 YLDP005. The socket for the bulbs should be E27, which is the standard socket type. Bulbholders should work between 100-240V. Bulbs must be able to operate over IEEE 802.11 s/g/n.

Software: In order for the program to be compiled, there must be a compiler that supports Python. The system has been tested on Visual Studio and Python IDEs. We assume compiler for necessary language is exists.

2.13. Requirement Specification

2.13.1. Interface Requirements

2.13.1.1. User Interfaces

The user interface language of our application will be English. It will be simple and easily understandable by users. In our application, there will be the part where the video will be played, the part where the functional keys will be found and a hidden part. Press the key to open the hidden section and the advanced settings tab will appear here. From the advanced settings tab, it will be possible to find, save, activate and test the bulbs options. The user will be able to make changes on the video and hardware through this interface.

2.13.1.2. Hardware Interfaces

The following table shows the hardware we use in our system. Explanations and visualizations have also been added.

Figure	Name	Definition & Usage Purpose
	Xiaomi Yeelight Smart Led Bulb 1s YLDP13YL	Smart light bulbs are LED light bulbs controllable using a smartphone, tablet, or smart home automation system. The bulbs will be used by AMP to reflect color codes coming through the network.[5]
	E27 Bulb Holder	E27 is the most common type of Edison Screw base, It is often referred to as just ES. The number '27' refers to the diameter. This type of base is also compatible with Compact Fluorescent Globes and LED globes. E27 is a 240 Volt bulb as is a standard Bayonet B22. We will use it to power our light bulb.[6]

	Linux Based Computer	A computer is an electronic device that manipulates information, or data. We will use the AMP software to process and send the data on the CPU.[7]

2.13.1.3. Software Interfaces

We will develop our project as a product. We use Python 3.7.0 when preparing and running the AMP software. While developing the software, we develop it through Visual Studio and Python IDLE. For interface design, we aim to use the PyCharm IDE's Qt Designer tool. Uploading the necessary libraries to the system will be provided by PIP. Since this application is open source, it can be developed/modified during the project.

2.13.1.4. Communication Interfaces

There is only one method that we used while communicating between our AMP and bulbs.

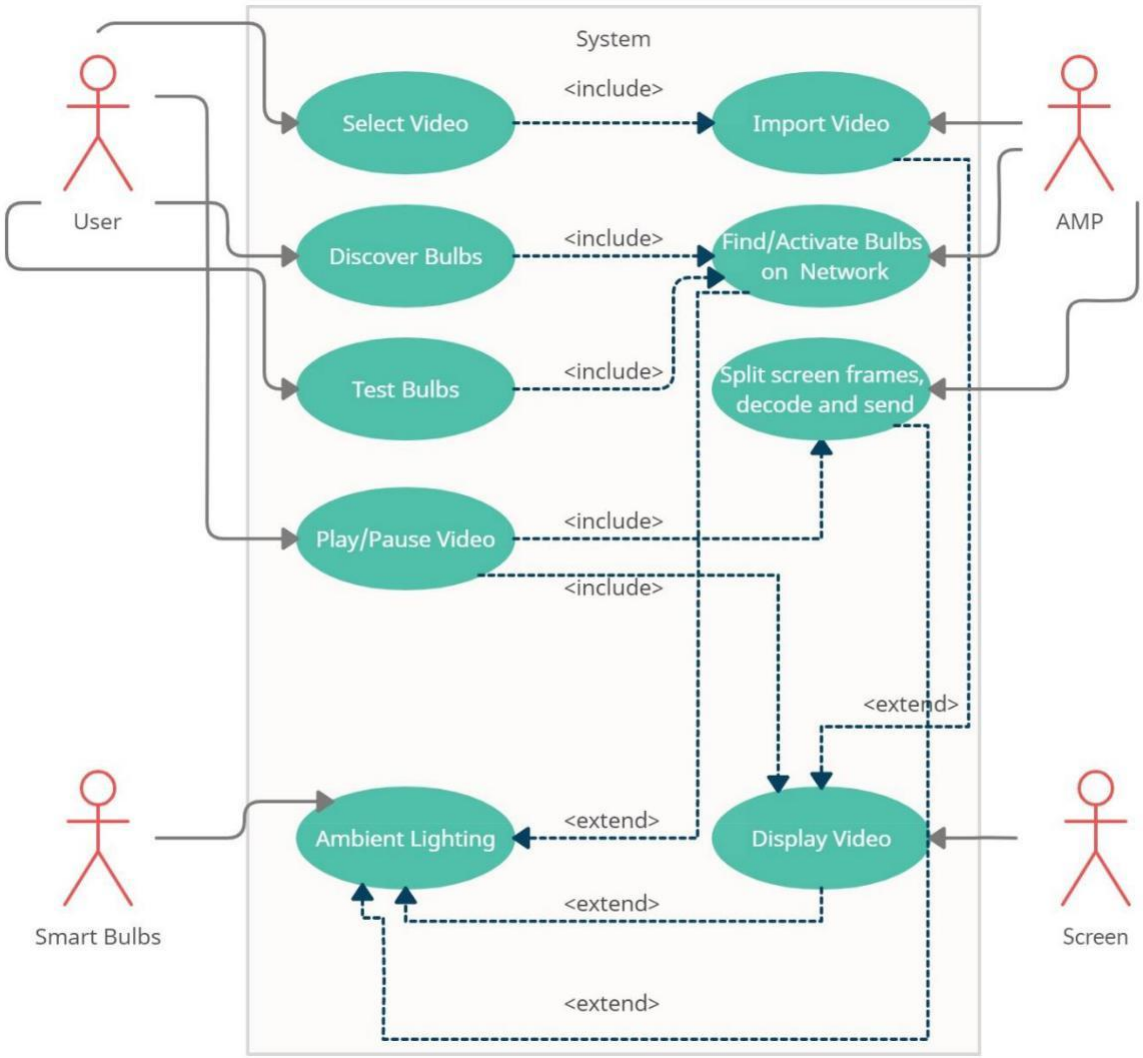
1) Wi-Fi

Using this technique, we inform bulbs when there is an data flow about color codes.

2.14. Detailed Description of Functional Requirements

In this part of the SRS, we stated functional requirements of the project.

2.14.1. Use Case Diagram



2.14.2. Use Cases

In below table, all use cases are explained briefly. Detailed explanation is given as subsections for each use case.

Use Case Title	Description
Select Video	The video to be played is selected by the User.
Import Video	The selected video is added to the system for playback by AMP.
Discover Bulbs	The user assigns the AMP to find the smart bulbs in his/her environment.
Find & Activate Bulbs on Network	AMP automatically finds the available smart bulbs in the network, saves them to the system and activates them.
Test Bulbs	The user tests the bulbs available in the system.
Play/Pause Video	Video can be started, paused and stopped by the user.
Split screen frames, decode and send	The colors of the video on the screen are decoded by AMP and transferred to the bulbs.
Ambient Lightning	The environment is illuminated according to the video colors.
Display Video	The video plays on the screen

2.14.2.1. Scenario 1: Select Video

Use Case Number	1
Use Case Name	Select Video
Summary	The video to be played is selected by the User.
Actor	User, AMP
Trigger	Open Video button is activated
Precondition	• AMP must be running on the operating system. • K-Lite Codec Pack must be installed on the operating system. • The file must be in video format.
Scenario	1. The user starts the program. 2. Selects the video you want to play from the drop-down menu and activates the upload button.
Exceptional Situations & Alternative Flows	1. Application may not work. • Restart the application.
Postcondition	1. AMP takes the video in its import queue.

2.14.2.2. Scenario 2: Import Video

Use Case Number	2
Use Case Name	Import Video
Summary	The selected video is added to the system for playback by AMP.
Actor	User, AMP
Trigger	Video is selected by user
Precondition	• AMP must be running on the operating system. • The video must have been selected by the user. • The video must be in a supportable format.
Scenario	1. The video selected by the user is queued. 2. If the selected video is in the correct format, it will be added to the video player inside the AMP software.
Exceptional Situations & Alternative Flows	1. Video may not open. • Upload a video in the correct format
Postcondition	1. AMP makes the video ready for playback and displays it on the screen.

2.14.2.3. Scenario 3: Discover Bulbs

Use Case Number	3
Use Case Name	Discover Bulbs
Summary	The user assigns the AMP to find the smart bulbs in his/her environment.
Actor	User, AMP
Trigger	The user directs the AMP to find bulbs on the network.
Precondition	• Smart bulbs must be identified on the network. • Bulbs must be connected to electricity and Wi-Fi.
Scenario	1. The user activates the Discover Bulbs button. 2. The AMP detects bulbs on the network via the Yeelight API. 3. The bulbs found on the network are registered in the program.
Exceptional Situations & Alternative Flows	1. No light bulbs may be found on the network. • Wi-Fi connection should be checked. • It should be checked whether the devices are defined in the network. • Network connection can be refreshed.
Postcondition	1. Light bulbs in the network will be found by AMP.

2.14.2.4. Scenario 4: Find & Activate Bulbs on Network

Use Case Number	4
Use Case Name	Find & Activate Bulbs on Network
Summary	AMP automatically finds the available smart bulbs in the network, saves them to the system and activates them.
Actor	AMP
Trigger	The Discover Bulbs button is activated by the user.
Precondition	• Smart bulbs must be identified on the network. • Bulbs must be connected to electricity and Wi-Fi. • The operating system must be connected to the network.
Scenario	1. When the button is activated, Xiaomi devices on the network are found via Yeelight API, smart bulbs are listed. 2. The listed bulbs are registered to the system in a certain order. 3. The bulbs whose registration process is completed become active.
Exceptional Situations & Alternative Flows	1. No light bulbs may be found on the network. • Wi-Fi connection should be checked. • It should be checked whether the devices are defined in the network. • Network connection can be refreshed.
Postcondition	1. Light bulbs in the network are registered in the program and activated.

2.14.2.5. Scenario 5: Test Bulbs

Use Case Number	5
Use Case Name	Test Bulbs
Summary	The user tests the bulbs available in the system.
Actor	User, AMP, Smart Bulbs
Trigger	The Test Bulbs button has been activated.
Precondition	• Bulbs must be defined and saved in the program. • Bulbs should be on. • Bulbs must be connected to Wi-Fi.
Scenario	1. The user activates the Test Bulbs button. 2. Cycle values that were previously prepared and embedded in the system are sent to the bulbs by the AMP. 3. The bulbs reflect the color codes they receive.
Exceptional Situations & Alternative Flows	1. Bulbs may not reflect color codes. • It is checked whether the bulbs are defined and active.
Postcondition	1. The bulbs reflect the color cycle sent.

2.14.2.6. Scenario 6: Play/Pause Video

Use Case Number	6
Use Case Name	Play/Pause Video
Summary	Video can be started, paused and stopped by the user.
Actor	User, AMP, Screen, Smart Bulbs
Trigger	The functional buttons of the AMP are activated.
Precondition	• The video must be selected. • Bulbs must be identified. • Light bulbs and Wi-Fi connections must be active.
Scenario	1. The selected video starts playing by the user. 2. The selected video can be paused by the user. 3. The selected video can be stopped by the user.
Exceptional Situations & Alternative Flows	1. The video may not start playing. • It should be checked that the file in the correct format is selected.
Postcondition	1. The video starts playing on the screen.

2.14.2.7. Scenario 7: Split screen frames, decode and send

Use Case Number	7
Use Case Name	Split screen frames, decode and send
Summary	The colors of the video on the screen are decoded by AMP and transferred to the bulbs.
Actor	AMP, Smart Bulbs, Screen
Trigger	The video has started playing.
Precondition	• The video must be selected. • Bulbs must be identified. • Light bulbs and Wi-Fi connections must be active.
Scenario	1. The video playing on the screen is decoded by AMP in the background. 2. The screen is divided into equal parts as much as the number of bulbs and the weighted colors are listed. 3. The codes of the listed colors are sent to the bulbs.
Exceptional Situations & Alternative Flows	1. Color codes may not go to bulbs. • It is checked whether the bulbs are defined and active.
Postcondition	1. The bulbs receive the color codes of the video that appears on the screen.

2.14.2.8. Scenario 8: Ambient Lightning

Use Case Number	8
Use Case Name	Ambient Lightning
Summary	The environment is illuminated according to the video colors.
Actor	AMP, Smart Bulbs, Screen
Trigger	Decoding and sending the color codes to the bulbs while the video is playing
Precondition	• The video must be selected. • Bulbs must be identified. • Light bulbs and Wi-Fi connections must be active. • Video must be playing.
Scenario	1. The bulbs analyze the codes that come to them and emit the right color led light.
Exceptional Situations & Alternative Flows	1. Bulbs may not reflect color codes. • It is checked whether the bulbs are defined and active.
Postcondition	1. The bulbs illuminate the environment simultaneously with the image on the screen in line with the color codes they receive.

2.14.2.9. Scenario 9: Display Video

Use Case Number	9
Use Case Name	Display Video
Summary	The video plays on the screen
Actor	User, AMP, Screen
Trigger	The video starts playing by the user.
Precondition	• The video must be uploaded to the system by the user.
Scenario	1. When the user presses the play button, the video starts playing.
Exceptional Situations & Alternative Flows	1. The video may not start playing. • It should be checked that the correct format video is selected.
Postcondition	1. The video starts playing on the screen.

2.15. Non-Functional Requirements

In this part of the report, we stated non-functional requirements of the project.

2.15.1. Performance Requirements

Performance Requirements	Description
Response Time	From the moment the video starts, there should be a delay of at most 1 second between the bulbs reflecting the colors. 1 second is a suitable time as there will be the phases of instantly buffering the screen, decoding the codes, sending it to the bulbs and finally getting a reaction from the bulbs. More than that will cause visual disturbance.
Error Handling	In case of an unexpected error, the software manufacturer should be contacted. Software support should be provided on an ongoing basis.
Scalability	AMP is controlled by the user through the operating system. The program will run stably unless an extra video is uploaded by the user while the program is running. This situation is blocked by the system.
Application requirements	The operating system where the program will run must have 100MB of free space. CPU speed or RAM of device is not a crucial concern.

2.15.2. Safety Requirements

Safety Requirements	Description
Installation Error	A properly working operating system must be present for installation.
Compilation Error	Compilation errors should be reported to the system.
Error Feedback	Error feedbacks should be reported to the developers through the system.
Network Protection	The network should be protected with an up-to-date password and there should be no leaks.

2.15.3. Security Requirements

Security Requirements	Description
Wi-fi Connection	The data on the AMP is regularly transferred to the bulbs via Wi-Fi. The Wi-Fi password must be protected to prevent leaks. The network password for the light bulb and computer must be set by the user and kept up to date.
Application Protection	Network and device information on the application is protected so that it cannot be exported. In case of any leakage, the user is responsible for what will happen.

2.15.4. Software Quality Attributes

Software Quality Attributes	Definition
Reliability	Under normal conditions, the program should run stably.
Portability	AMP software should work on Linux and Windows.
Correctness	The system should transmit color codes at a uniform rate and without delay.
Learnability	System shall be easy to understand and simple to use for ordinary customers.
Maintainability	In case of any error, the system should compile itself and reach the solution.
Extensibility	The system should be in an expandable state with new functions to be added..
Testability	The system should work without errors and all cases where an error may occur should be testable.
Efficiency	The system should be able to run smoothly in relation to the CPU and Wi-Fi connection quality.
Usability	The product should be easily usable and discoverable by users.
Administrability	The connection information of the bulbs should also be entered into the system manually.

3. Software Design Description (SDD)

3.1. Introduction

The purpose of this Software Design Document is to provide details and what kind of system software Ambilight Media Player should include. This document includes the working principles of the methods that we will develop. The target audience of this project is users who want a cinema environment in their home. It aims to bring cinema ambiance and realism to your home by using smart bulbs. It is aimed to create an environment containing real world objects. This program mainly contains a single mode, but there are many situations that will vary according to the number of lamps. Although the system will make the definitions automatically, it will also offer the opportunity to make adjustments manually. Python software language and Yeelight API will be used in the system.The K-Means library will be used to calculate the color intensity and convert it to vectorial values. In the future process of our development, these techniques can be changed based on test results we received. Therefore, we will decide step by step as we receive outputs of our system.

3.2. Purpose of This Document

The technical design for Ambilight Media Player is described in this Software Design Document. The primary goal of this document is to describe the technical vision for how the SRS document's requirements will be met. It will also give a general overview of the system's design architecture. As a result, this document covers all of the interfaces, diagrams, and interactions that we used during the project's development.

3.3. Scope of the Project

With this project, it is aimed to design and develop software that will enable users to maximize their cinema experience and provide an experience that will make them feel close to reality. The following functionalities are included in our project:

Video switching and video playback
Adding bulbs to the system either automatically or manually is possible.
The ability to activate or deactivate the inputted lamps.
Ability to simultaneously buffer an opened movie, decompose its colors, and transmit it to the appropriate lamp Ambient lighting.

3.4. Glossary (Definitions, Acronyms, and Abbreviations)

TERM	DEFINITION
Ambilight	The TV's edges have lamps that light up the wall in the same colors as the screen. This makes the screen appear larger while also making watching TV in the dark less straining on the eyes.
SDD	It is the document that explains the software project's design.
UML Diagram	It is a modeling language that is used in software engineering to visualize the structure of a system.
Python	Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis.
Yeelight / API	Yeelight Smart LED Bulb is an affordable color LED bulb that user can control. Yeelight Python library is a small library that lets user control bulbs over Wi-Fi.
K-Means	Kmeans is an Iterative algorithm that divides a group datasets into subgroups based on the similarity and their mean distance from the centroid of that particular subgroup.
Scrum	Scrum is a software engineering application development method.
Management of Risk	During the construction phase of a project, it is a way of classifying, evaluating, and managing dangers to the project.

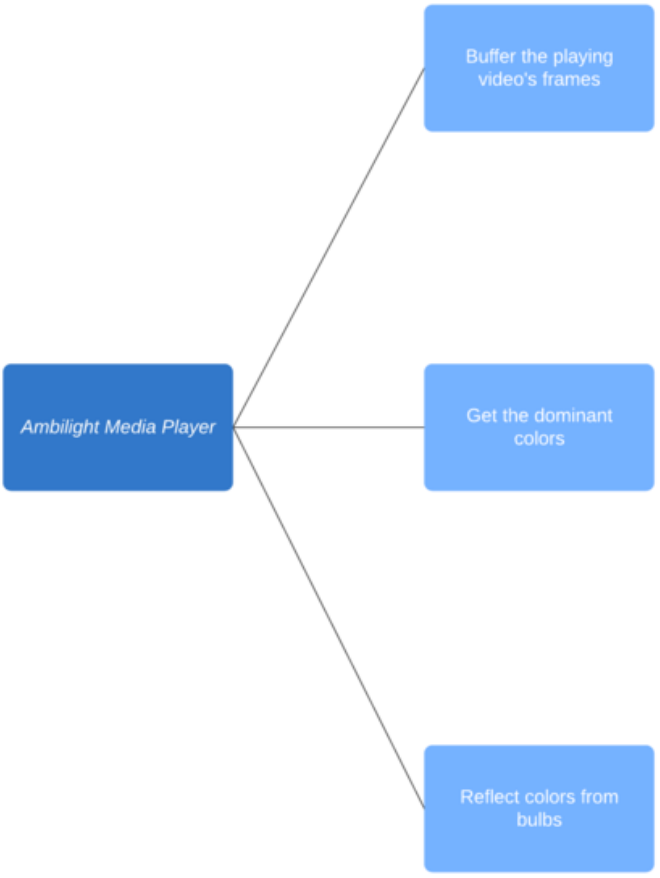
3.5. Overview of the Document

The SDD is divided into partitions, and the contents of each partition are listed below. Part 2 is the systemdesign section, which describes the diagrams that comprise the system and architectural design of the project stages, displays design features, and explains the diagrams that comprise the system's exceptions and conditions. The third part User Interface Design. Part 4 is Use Case Realization. In this section, the block diagram of the system is drawn and briefly explained. The last part is about the Environment.

3.6. System Design

3.6.1. Architectural Design

In this section, we described the problem description, technologies used and diagrams of our architectural design. AMP system will include 3 main tasks. Figure-1 below shows the architectural design of AMP.



3.6.2. Problem Description

Eye strain occurs when people spend watching something on the television or computer daily, and these cause various discomforts.[13] In a home cinema room lighting condition Ambilight can reduce eye strain in 60-90% of the people dependent on the Ambilight setting. Viewing with Ambilight is experienced by most viewers as more pleasant and comfortable for the eyes than without it.[14] Whether you're gaming, watching your favorite programs or listening to music, Ambilight Media Player is designed to create an immersive entertainment experience with intelligent bulbs.

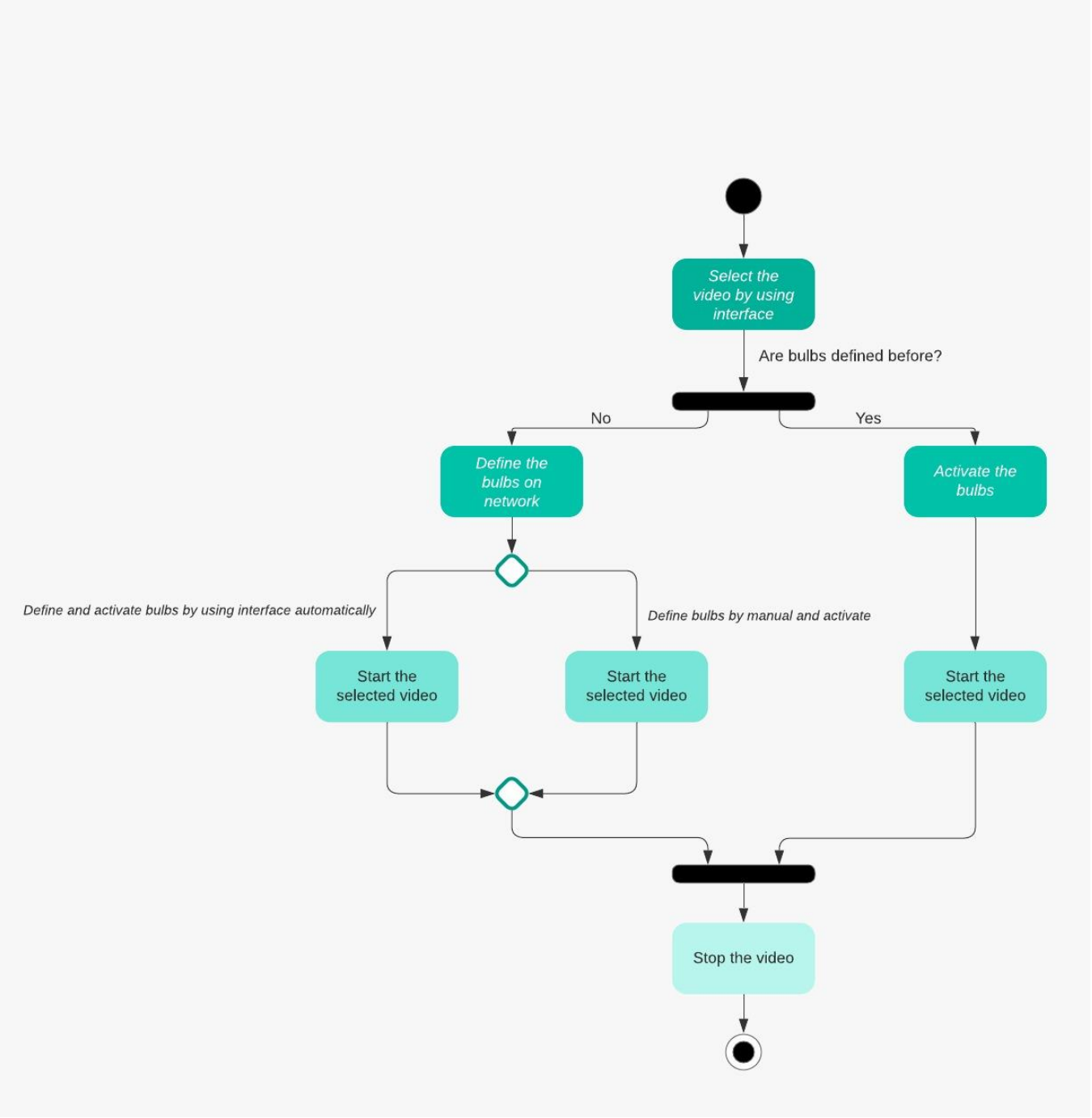
3.6.3. Technologies Used

All of the hardware products we use in our system are listed below:

- LG 47LA620S Full HD (FHD) TV
- Abra A7 V13.1 17,3" Laptop
- 4 piece of Xiaomi Yeelight 1s YLDP13YL Rgbw Akıllı Led Ampül (3rd GEN)
- UGREEN HDMI Cable 4K 6FT Braided High Speed HDMI Cord 18Gbps
- Lamp Holders
- Hotspot providing 2.4Ghz internet Access

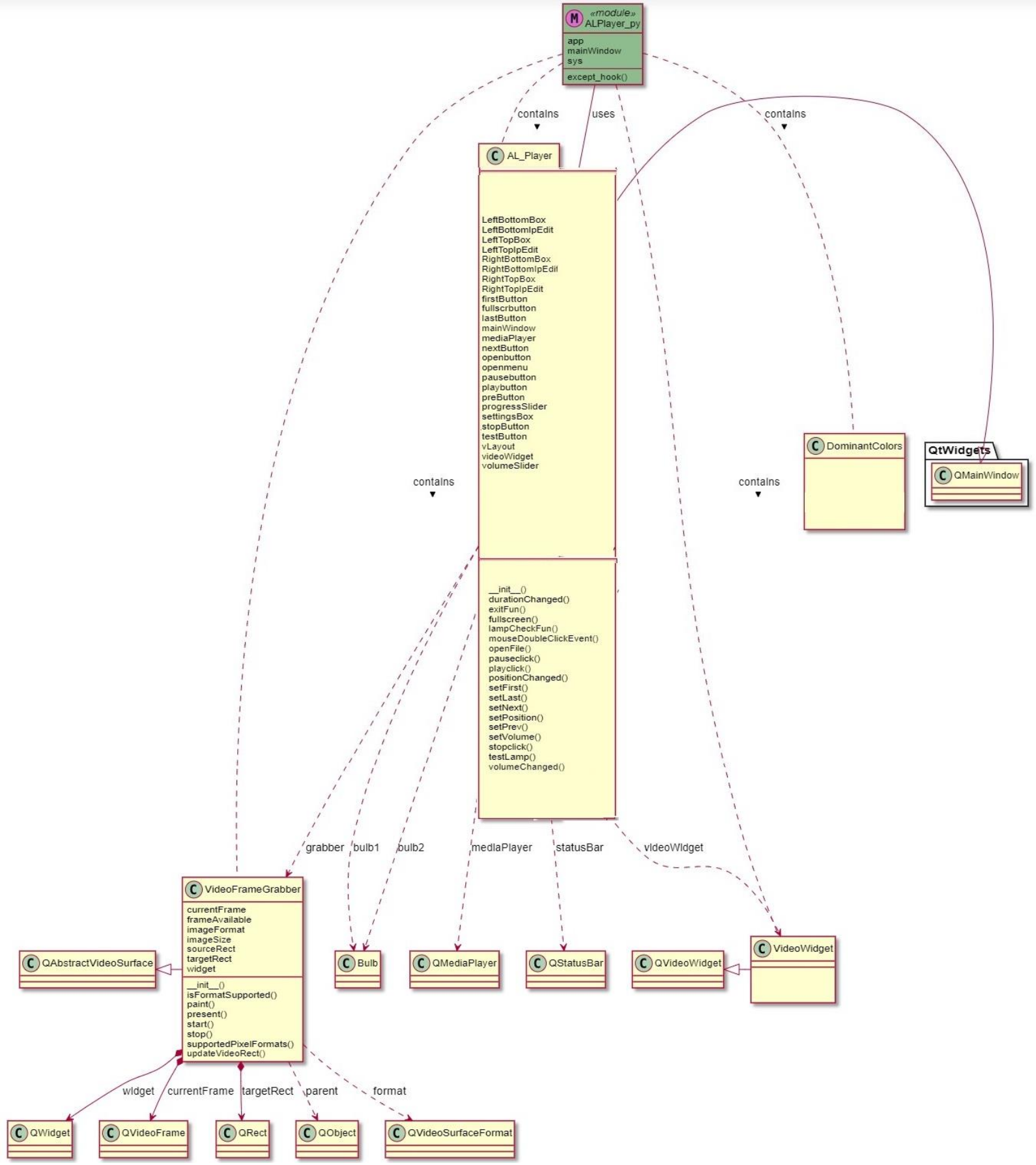
3.7. Activity Diagram

Figure below shows the activity diagram of AMP



3.8. Class Diagram

Figure below shows the class diagram of AMP.

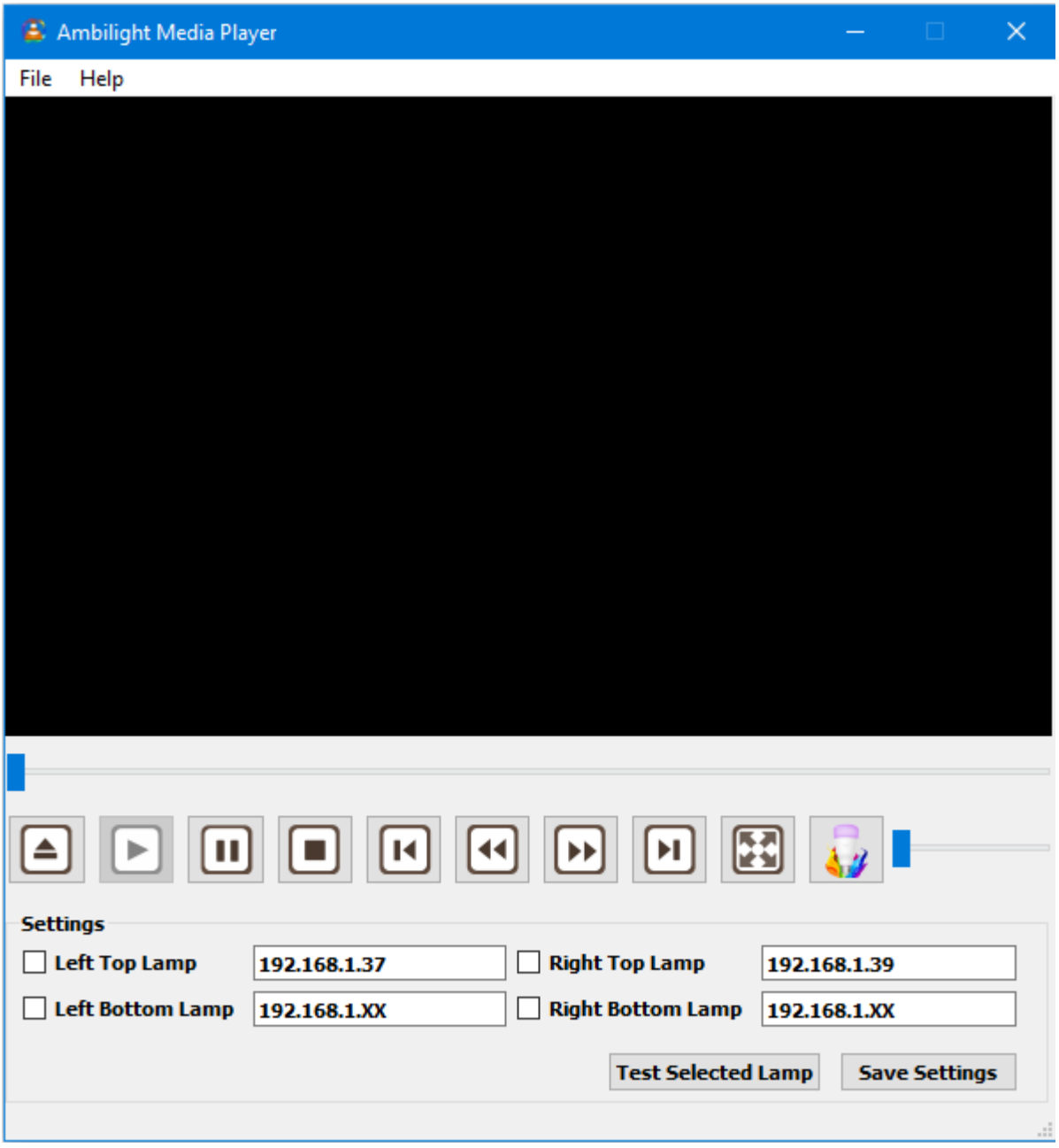


3.9. User Interface Design

In this section, we'll describe our designs for various use cases and what features each UI screen includes.

3.9.1. Main Menu

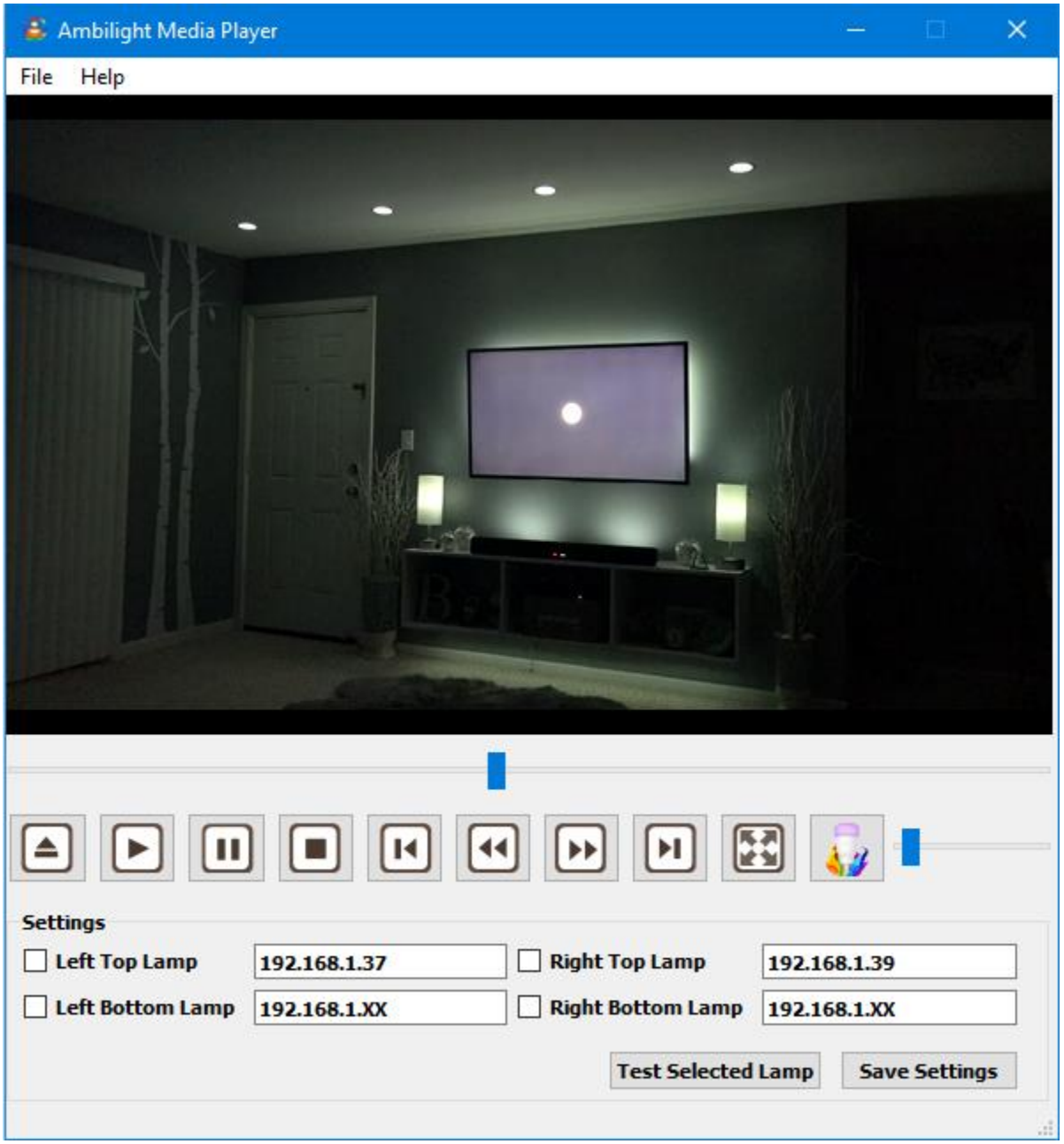
Figure below shows the main menu of AMP.



In order to run any feature of AMP, the user must first select a video. For this reason, a video that is already in the system should be selected by using the open video button. When one of these options is selected, a video will appear on the screen to be played.

3.9.2. Video-Play Menu

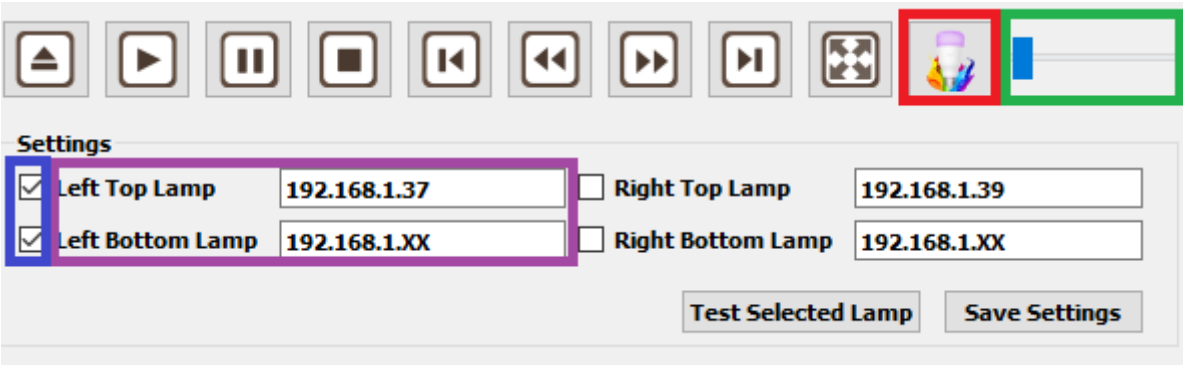
Figure below shows the playing video of AMP.



After the video starts to play, the threads can be automatically searched on the network and discovered on the network thanks to the Yeelight API library.[15] Or manually, IP addresses can be entered and activated automatically.[16]

3.9.3. Bulb Connection Toggle Menu

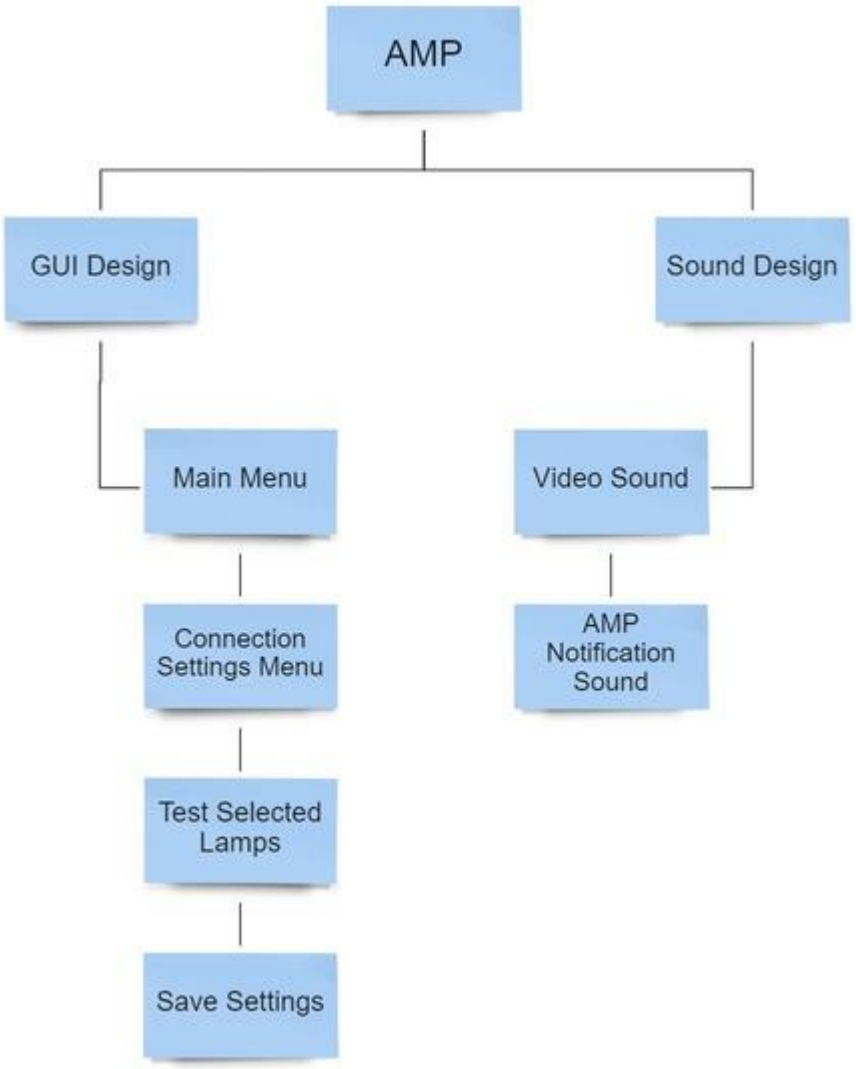
Figure-5 below shows the hidden connection settings menu of AMP.



The user can open the lamp connection settings menu, which is turned off, using the red button. You can adjust the sound settings from the button shown in green. He can activate whichever lamps he wants from the part shown in blue. It can enter the IP addresses of the lamps manually or automatically from the section shown in purple. After recording them, he can enjoy the environment by starting the video.[15]

3.10. Use Case Realization

3.10.1. Brief Description of Figure



The components of the AMP Project are shown above in Figure 6. All designed systems of the program are shown in the block diagram in the figure. The subsystems of the system are divided into two subcomponents.

3.10.1.1. Graphical User Interface

The GUI design explains the relationship between the actors and the system. There are four subsystems in this design: Main Menu, Connection Settings Menu, Test Selected Lamps and Save Settings. The Main Menu is a start page. The Settings Menu contains options for the IP adresses and activate buttons of bulbs, you can change and show the settings by clicking red button shown in Figure-5. Test Selected Lamps will send random color query to bulbs and they will reflect these colors. Save settings button will save the datas of bulbs and users so next time they will not be needed again.

3.10.1.2. Sound Design

Sound design can be evaluated in two different classes. As Video Sound and Notification Sound. Video Sounds are sounds that vary depending on the video being played. Notification Sounds are the response sounds of the program when a certain action is taken from the program.

3.11. Conclusion & Discussion

For the CENG407 coded graduation project, we explained the documentation process of the Ambilight Media Player project in detail in our report. The aim of this study is to create a product that works with smart lamps, is affordable and optimized, and will provide a good Ambilight environment with its synchronization. In the Literature Review, we searched for which algorithms, software languages and libraries we should use for our project. According to the results of our research, we will use Python, Yeelight API, K-Means, OpenCV, PyQt5. In our SRS document, we considered the requirements of our system and explained the necessary actions. We've identified use cases and schematics that show how the system will work. In our SDD document, we have determined the architecture of our interface, where we will perform the process of receiving, buffering, coloring and sending the image to the lamps over the network. In addition, we aimed to make our project more understandable and traceable by drawing diagrams showing our software architecture.

4. References

[1] About Python. (2021, November 9). Python.Org. Retrieved November 1, 2021, from https://www.python.org/about/

[2] What Is Python Used For? A Beginner's Guide. (2021, September 22). Coursera. Retrieved November 1, 2021, from https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python

[3] YeeLight library. (2020, June 17). Yeelight. Retrieved November 2, 2021, from https://yeelight.readthedocs.io/en/latest/

[4] Sharma, P. (2021, April 26). K Means Clustering Simplified in Python. Analytics Vidhya. Retrieved November 9, 2021, from https://www.analyticsvidhya.com/blog/2021/04/k-means-clustering-simplified-in-python/

[5] Real Python. (2021, January 8). K-Means Clustering in Python: A Practical Guide. Retrieved November 9, 2021, from https://realpython.com/k-means-clustering-python/

[6] Choudhury, A. (2021, May 29). 8 Python GUI Frameworks For Developers. Analytics India Magazine. Retrieved November 10, 2021, from https://analyticsindiamag.com/8-python-gui-frameworks-for-developers/

[7] Topcoder Opencv Library. (2021, April 2). Topcode. Retrieved November 10, 2021, from https://www.topcoder.com/thrive/articles/what-is-the-opencv-library-and-why-do-we-need-to-know-about-it

[8] What Is the Flutter Framework? (2021, May 18). Perfecto by Perforce. Retrieved December 17, 2021, from https://www.perfecto.io/blog/what-is-flutter-framework

[9] Coolblue. (2021, September 21). This is the Philips Ambilight - Coolblue. Retrieved November 7, 2021, from https://www.coolblue.nl/en/advice/philips-ambilight.html#id-what-is-philips-ambilight

[10] Lightberry. (2013, January 24). Lightberry. Retrieved November 8, 2021, from http://guide.lightberry.eu/

[11] DreamScreen HD & 4K - Smart TV Backlighting. (2018, November 21). Kickstarter. Retrieved November 9, 2021, from https://www.kickstarter.com/projects/237916976/dreamscreen-hd-and-4k-led-backlighting-for-any-hdm/description

[12] Choudhury, A. (2021, May 29). 8 Python GUI Frameworks For Developers. Analytics India Magazine. Retrieved November 10, 2021, from https://analyticsindiamag.com/8-python-gui-frameworks-for-developers/

[13] Topcoder Opencv Library. (2021, April 2). Topcode. Retrieved November 10, 2021, from https://www.topcoder.com/thrive/articles/what-is-the-opencv-library-and-why-do-we-need-to-know-about-it

[14] YeeLight library. (2020, June 17). Yeelight. Retrieved November 2, 2021, from https://yeelight.readthedocs.io/en/latest/

[15] Yeelight Smart LED Light Panels [Preorder]. (2021, December 9). Yeelight Shop. Retrieved December 9, 2021, from https://us.yeelight.com/shop/yeelight-smart-led-light-panels/

[16] Need An Extra Hand? Maybe Smart Light Bulbs In Your Home Can Help. (2021, September 16). Lifewire. Retrieved December 9, 2021, from https://www.lifewire.com/smart-light-bulbs-4149561