CONESTOGA COLLEGE - SET

ARc-Light SRS

Project Design | PROG2030

Date of Completion: 2020-03-31

TEAM AgTech: Kyle Horsley, Bence Karner, Randy Lefevbre, Lucas Roes

|  |  |  |  |
| --- | --- | --- | --- |
| Rev | Date | Description Of Change | Completed By |
| 1.0 | 10/23/2019 | Initial Document Creation | Bence Karner |
| 1.1 | 10/24/2019 | Contents for Aisle Mapper Added | Bence Karner |
| 2.0 | 12/02/2019 | Project Changed to Electrical AR | Kyle Horsley |
| 2.1 | 12/04/2019 | Contents changed to new Capstone project | Kyle Horsley |
| 2.2 | 01/31/2020 | AR Development tool and tool references changed from Vuforia to AR Foundation | Lucas Roes |
| 2.3 | 2/28/2020 | Features Added: Section 2.2, 3.2 | Kyle Horsley |
| 2.4 | 3/30/2020 | Updated sections:  1.3 – Added definitions for ARFoundation, and ARCore  1.4 – Added reference to the ARCore documentation  2.1 – Updated 2.1.2.3 from Light Adjustment Options Panel, to Light Adjustment Radial Menu  2.2 – Corrected an error with incorrectly numbered sub sections. Previously defined features with designations in the range of 2.2.7 – 2.2.10 have been updated to 2.2.11 – 2.2.17. Added new features 2.2.7 – 2.2.10  2.7 – Removed 2.7.3 as the system is not collecting data, nor communicating with external sources  3.2 – Updated all of 3.2 with minor clarifications. Added 3.2.8 -3.2.17 for the updated features from section 2.2  3.3 – Removed 3.3.1 System Response Time, and 3.3.2 User Capacity  3.4 – Removed specifications from section 3.4 as they’re no longer relevant  3.5 - Removed 3.5.3 Network Connection  Appendix – Updated figures 1 – 5 with new UI mockups. Also added new UI mockups for the additional features added in section 2.2. Added/updated all figures in Appendix B with improved use case diagrams | Bence Karner |

# Revision History

Contents

[Revision History 1](#_Toc36584373)

[1.0 Introduction 3](#_Toc36584374)

[1.1 Purpose 4](#_Toc36584375)

[1.2 Scope 5](#_Toc36584376)

[1.3 Definitions 6](#_Toc36584377)

[1.4 References 7](#_Toc36584378)

[1.5 Overview 8](#_Toc36584379)

[2.0 Overall Description 9](#_Toc36584380)

[2.1 Product Perspective 10](#_Toc36584381)

[2.2 Product Functions 12](#_Toc36584382)

[2.3 User Characteristics 14](#_Toc36584383)

[2.4 Constraints 15](#_Toc36584384)

[2.5 Assumptions and Dependencies 16](#_Toc36584385)

[2.6 Apportioning of Requirements 17](#_Toc36584386)

[2.7 System Attributes 18](#_Toc36584387)

[3.0 Specific Requirements 19](#_Toc36584388)

[3.1 External Interface Requirements 19](#_Toc36584389)

[3.2 Functional Requirements 19](#_Toc36584390)

[3.3 Performance Requirements 24](#_Toc36584391)

[3.4 Logical Database Requirements 25](#_Toc36584392)

[3.5 Design Constraints 26](#_Toc36584393)

[Appendix A: 27](#_Toc36584394)

[Appendix B: 35](#_Toc36584395)

# 1.0 Introduction

This document outlines the features and functionality that will come with the 2019 Arc-Light capstone project. This project will be developed by April 21, 2020 and will use this as a guideline of how, when and where things should be happening. This document is split into sections based on the user’s technical background and provides enough information for all users to understand the direction of this project moving forward. This software requirements specification will greatly help the development of the augmented reality capstone project.

1.1 Purpose

The purpose of the document is to provide a detailed description of the requirements, and specifications for the proposed system. Sections one and two is intended for non-technical users, project staff and stakeholders. Individuals with a background in development, engineering, or those who wish to better understand the technical nature of the system, are encouraged to read section three, where the systems specifics are examined in greater detail.

1.2 Scope

The proposed ARc-Light project will provide electricians and home renovators an easier method to visualize their lighting layout and illumination in a given environment. This augmented reality mobile application will be developed for android users. Built in Unity using AR Foundation, this application will combine the fun of augmented reality with the practical use of electrical and lighting work.

This application will allow users to place lighting fixtures within a room and illuminate it with a selection of bulbs, lumens and other various settings. The application will include lamps, wall fixtures, chandeliers and ceiling lighting. Each one of these fixtures will illuminate differently and the goal is to allow users to be sure about their lighting selection. This application will be able to layout the most appropriate lighting layout for a room based on the dimensions given and the lighting option chosen.

Using this application will make electricians jobs easier and relieve some stress from homeowners. The goal of this project is to create a successful working augmented reality project that will allow users to place lighting within their home and illuminate the room, using the selected lighting, to ensure they are making the right choice.

1.3 Definitions

|  |  |  |
| --- | --- | --- |
| Name | Report Notation | Description |
| User | User | The individual directly interacting with the system |
| AR | System | Augmented Reality |
| Android Application | Mobile App | The android version of the mobile application |
| Unity | System | The computer program used for development |
| App | Slang | Meaning application. The product we are building. |
| SRS | Document | Software Requirements Specification |
| Interface | System | A point where two systems, subjects, organizations, etc., meet and interact. Or A device or program enabling a user to communicate with a computer. |
| API | System | In computer programming, an application programming interface (API) is a set of subroutine definitions, protocols, and tools for building application software. |
| GB | System | Gigabyte. A unit of measurement for digital storage. Equal to a thousand bytes. |
| RAM | System | Random Access Memory. The main form of memory in a computing device. Random Access Memory is volatile. That means data is retained in RAM as long as the computer is on, but it is lost when the computer is turned off. When the computer is rebooted, the OS and other files are reloaded into RAM, usually from an HDD or SSD. |
| ARFoundation | System | Package for Unity that allows for cross platform development of AR applications. |
| ARCore | System | As per the official documentation, “ARCore is Google’s platform for building augmented reality experiences.” (ARCore, ND). |

1.4 References

ARCore. (ND). ARCore overview. Retrieved March 30, 2020, from https://developers.google.com/ar/discover

IEEE Standards Association. (1998). *IEEE Guide for Software Requirements Specifications*. Retrieved on Oct 31, 2019, from https://conestoga.desire2learn.com/

content/enforced/62087-D2LGroup-0059\_P00\_X\_X\_D2LGroup-0059\_1\_X/Standards

/SETStandards.html

Google. (N.D. a). *Codenames, Tags, and Build Numbers*. Retrieved November 30, 2019, from https://developers.google.com/ar/develop/java/quickstart

Google. (N.D. b). *Quickstart for Android*. Retrieved November 30, 2019, from https://source.  
 android.com/setup/start/build-numbers

Karner, B., Lefebvre, R., Prince, K. (2018). GIORP-Team-04: SAaD: ML-04 - The GIORP SRS. Retrieved on October 31, 2019, from https://conestoga.desire2learn.com

/d2l/lms/dropbox/user/folders\_history.d2l?db=190792&grpid=190671&isprv=0&bp=0& ou=185223

Luminaire. (N.D.). The Augmented Reality App for Lighting Business. Retrieved November 30, ` 2019, from https://luminaire-ar.com/en/the-app/

1.5 Overview

The next section, Overall Description, gives an overview of the functionality of the system. It describes the interfaces the system will have to communicate with external actors and systems. This section also summarizes the major functions of the system, as well as explains the expectations the system requires of the user. Lastly, it outlines the constraints of the system.

The third section, Specific Requirements, is written primarily for the developers and describes in more technical terms and in greater detail the interfaces, functionality, requirements, and constraints of the system. Also, it describes the logical database requirements.

The same system is being described in both sections; the difference is the level of technicality used to describe the system. The second section of this document is meant to be read by general stakeholders or customers, where the third is generally for developers working on the system.

# 2.0 Overall Description

The material covered in the following section is intended to provide a clear depiction of the systems features, settings, and potential constraints. The section is intended for readers who desire a complete high-level understanding of the product.

The material covered in the following sub sections is provided to give the reader a comprehensive background of the system in order to understand the specific requirements as outlined in section three.

2.1 Product Perspective

Competing applications are currently available for both iOS, and Android devices. The four most notable examples include Luminaire, Zoom-AR, IKEA Place, and Wayfair.

Luminaire can be described as an application designed for dividing realistic lighting simulations are targeted towards and commercial entities (Luminaire, N.D.). The application provides users with the option to simulating certain lighting fixtures in both interior and exterior environments and present these in a professional manner using a digital showroom.

Zoom-AR is an Android mobile application available to interior designers and other business professionals. The application gives uses a great deal of freedom in defining how their selected light fixture appears on the device. For example, users can select specific lighting model and apply different color scheme to change its appearance. It also allows professionals to add new 3-D models to the applications library and brand their creations.

The IKEA Place, and Wayfair app help users to select from a catalog of available light fixtures and simulate their appearance in their current environment using augmented reality. In both cases, there app uses Apples, augmented reality framework on IOS devices to allow users to scan their environment, simulate their selected furniture to scale, and share the resulting view with their contacts.

From a technical perspective many of the currently available applications fulfill much of the same requirements outlined for the proposed system, with a few notable exceptions. Where both Luminaire and Zoom-AR provide the user the ability to customize and render a light fixture in 3D, they don’t however, fully simulate the light source, and illuminate the room as if the lights were present. Both applications are also geared more towards commercial/business use, with a myriad of additional features not required by our target audience. Both IKEA Place, and Wayfair provide much of the same functionality as the previous examples but are more focused on assisting the end consumer. In both these cases, while their catalogue of selectable light fixtures is large, the user’s capacity to interact with the lights and change their visual appearance is limited.

2.1.1 System Interfaces: The system will not require the user of external interfaces to perform its functions

2.1.2 User Interfaces:Users will interact with the mobile app via the touch screen using the following interfaces

**2.1.2.1: The Application Main Screen**

Using the applications main screen, users will be able to see their environment through the camera module. Additionally, they will be able to interact with the camera, and use it to capture images of their surroundings. See figure 1 in the appendix for a mockup of the UI

**2.1.2.2: The Slide Out Menu**

The slide out menu will give users the ability to select from a list of different light types. Each light will be represented with a unique name, and image of the 3D model used to represent it in the AR view.See figure 2 in Appendix A for a mockup of the UI

**2.1.2.3: Light Adjustment Radial Menu**

While in the applications main screen, users will be able to adjust the output of the light fixture’s properties using the radial menu. The lights range, color, brightness, orientation, camera effects, can be altered using this menu. See figures 3, 4, 5, 6, 7, and 8 in Appendix A for a mockup of the UIs

2.1.3 Hardware Interfaces: The system does not communicate with, nor control specific hardware components.

2.1.4 Software Interfaces: The system does not communicate with, nor require the use of specific software interfaces.

2.1.5 Communication Interfaces: All communications will be restricted to user’s device and will not require the use of specific underlying communications protocols.

2.1.6 Memory Constraints:Specific memory constraints have not been defined and are not relevant to the systems design or requirements.

2.1.7 Operation:The system will support a single operational mode and allow the user to interact will all its features regardless of the individual using the software.

2.1.8 Site Adaptation Requirements:The system does not require site specific adaptational changes.

2.2 Product Functions

The mobile app will support the following features and functions:

**2.2.1 View a room image:** The system shall allow the user to use the device camera to capture an image of the room that they wish to alter.

**2.2.2 Get list of lights:** The system shall get a list of available light options for the user to select from. See figure 9 for the corresponding use case diagram.

**2.2.3 Select light source:** The system shall allow the user to select from the list of light sources based on their type.

**2.2.4 Place lighting:** The system shall allow the user to place light fixtures in the digitally captured room. Users will be able to use default light fixtures, as well as create custom lights with specific characteristics. See figure 9 for the corresponding use case diagram.

**2.2.5 View result:** The system shall allow the user to view a simulation of the room with the user’s customizations, accurately rendering the new lighting. See figure 9 for the corresponding use case diagram.

**2.2.6 Save lighting alterations:** The system shall allow the user to save the new lighting configuration to their device.

**2.2.7 Share image:** The system shall allow the user to take a screenshot of the AR room and can share the image, as well as save the image to their device storage. See figure 9 for the corresponding use case diagram.

**2.2.8 Adjust brightness:** The system shall allow the user to adjust the brightness of an induvial light. See figure 9 for the corresponding use case diagram.

**2.2.9 Adjust light color:** The system shall allow the user to adjust the lights color. See figure 9 for the corresponding use case diagram.

**2.2.10 Adjust light range:** The system shall allow the user to adjust the light range, that is, the total distance from the light source where the light still provides illumination. See figure 9 for the corresponding use case diagram.

**2.2.11 Adjust lighting effects:** The system shall allow the user to adjust the post processing effects applied to the lights (e.g. ambient occlusion, and bloom). See figure 9 for the corresponding use case diagram.

**2.2.12 Toggle light on and off:** The system shall allow the user to toggle a light on, or off

See figure 9 for the corresponding use case diagram.

**2.2.13 Delete item:** The system shall allow a user to delete an already placed item

See figure 9 for the corresponding use case diagram.

**2.2.14 Move item:** The system shall allow a user to move an already placed item. See figure 9 for the corresponding use case diagram.

**2.2.15 Rotate item:** The system shall allow a user to rotate the item once placed. See figure 9 for the corresponding use case diagram.

**2.2.16 Place multiple items:** The system shall allow the user to place multiple items in different locations throughout the room.

**2.2.17 Save light location:** The system shall save the lights location and allow the ability to scan the whole room and come back to it while the lights remain in the same place.

2.3 User Characteristics

The target user for the mobile app are electricians and home renovators. These people will be more likely to understand how the different lighting systems will affect a home’s look and feel. However, this application will not require technical training, or specific education related to its use. Users will require an understanding of the English language, and how to operate their mobile device. The user should be capable of understanding augmented reality and how it works within the mobile world. The application could be set up for use by any individual and no training will need to be completed in order to use this application.

2.4 Constraints

In order to build the application to an Android device, ARFoundation requires the use of ARCore. As per the ARCore specifications, the development toolset will be restricted to Android Studio version 3.1 or higher, with Android SDK Platform version 7.0 (API level 24 Nougat) or higher. Development will occur on API level 27 (i.e. Oreo version 8.1.0) using the default Android Emulator (version 27.2.9 or later) as part of Android studio (Google, N.D. a) (Google, N.D. b).

2.5 Assumptions and Dependencies

Some assumptions have been made for the purpose of completing the initial draft of the specifications. They are as follows.

**2.5.1 Development Tools**:  It is assumed that the development tools, Android Studio, Unity, and AR Foundation, will be freely available to use.

**2.5.2 Camera Quality:** It is assumed that the device the system runs on will have a functional camera of high enough quality to capture an image of the room.

2.6 Apportioning of Requirements

Future versions of the system will see the android application ported to be compatible with iOS-based devices. Additionally, a suit of additional web-based services is planned to provide business owners with greater control over the captured data and provide additional options for analyzing the data.

2.7 System Attributes

The following section documents the non-functional requirements of the system. These have been selected and detailed below, in order to ensure all aspects of the systems objectives can be verified as complete.

**2.7.1 Reliability:** The system shall work with 99% reliability before release.

**2.7.2 Availability:** At this time, the system has been specifically designed to operate on Android mobile devices with Android 7.0 or higher.

**2.7.3 Security:** Because the system is not collecting user data, nor is it communicating with systems external to the device, no security attributes have been defined.

**2.7.4 Maintainability:** At this time, the system is expected to be supported until our capstone demonstration has been completed, in April 2020. If an industry partner is acquired, this date will be discussed and revised with our partner.

**2.7.5 Portability:** At this time, the system has been specifically designed to operate on Android mobile devices with Android 7.0 or higher. Portability to IOS is a tentative goal for the future but will have no impact regarding current design choices.

# 3.0 Specific Requirements

The following sub-sections re-examine the material covered in the prior sections, with the intent of providing additional details regarding the systems functional requirements. Each of the planned features are examined in greater detail and include any additional quality requirements that must be surpassed before they are considered as complete.

3.1 External Interface Requirements

At this point in time, no external interfaces have been defined for the system.

3.2 Functional Requirements

**3.2.1 Functional Requirement 2.2.1**

|  |  |
| --- | --- |
| Description | The system shall scan the room and detect its three-dimensional characteristics |
| Sequence of Operations | The user will activate the application from their mobile device, and point the camera at the area where they wish to place a light fixture |
| Exceptional Response | Disallow the user from placing a light fixture on non-valid planes |
| Inputs | Application startup process |
| Outputs | Visual feedback from the camera module |
| Reference | See figure 9 in Appendix B |

**3.2.2 Functional Requirement 2.2.2**

|  |  |
| --- | --- |
| Description | The system shall retrieve a list of lighting fixtures from the devices memory |
| Sequence of Operations | The user will activate the slide out menu select from the list of available items |
| Exceptional Handling | Display error message, and disallow the user from selecting and placing lights |
| Inputs | Button press to enable the light selection panel and make it visible |
| Outputs | Appearance of the slide out menu with a populated list of lighting sources |
| Reference | See figure 9 in Appendix B |

**3.2.3 Functional Requirement 2.2.3**

|  |  |
| --- | --- |
| Description | The system shall allow the user to select a light fixture from the list of available options |
| Sequence of Operations | The user will slide out the light selection sub view, and tap on the light source they wish to select |
| Exceptional Handling | Display error message, and disallow the user from continuing to the next step |
| Inputs | Tap gesture recognizer with the name of the selected light fixture |
| Outputs | Appearance of the selected light source in the AR view |
| Reference | See figure 9 in Appendix B |

**3.2.4 Functional Requirement 2.2.4**

|  |  |
| --- | --- |
| Description | The system shall apply the selected light model to the environment/room |
| Sequence of Operations | The user will select a model from the list of available lights, and apply the fixture to a section of the view |
| Exceptional Handling | Display error message, and disallow the user from placing the light |
| Inputs | The selected light model, and the location of the user’s touch |
| Outputs | Visual representation of the selected light fixture at the location set by the user |
| Reference | See figure 9 in Appendix B |

**3.2.5 Functional Requirement 2.2.5**

|  |  |
| --- | --- |
| Description | The system shall simulate a realistic lighting effect from the AR light |
| Sequence of Operations | The system shall recognize the characteristics and features of the environment in view, and apply their lighting effect in such a way that the illumination pattern resembles a life like light source |
| Exceptional Handling | Display error message, and disallow the user from continuing to the next step |
| Inputs | Light model selected by the user, and its location |
| Outputs | NA |
| Reference | See figure 9 in Appendix B |

**3.2.6 Functional Requirement 2.2.6**

|  |  |
| --- | --- |
| Description | The system shall automatically save the state and properties of the instantiated lights |
| Sequence of Operations | The user will select a light source, set its properties, and allow the system to save the updated values after each action |
| Exceptional Handling | Display error message, but allow the user to continue changing the light source and its settings |
| Inputs | Selected light fixture, and its properties |
| Outputs | Saved details defining the light source and its settings |
| Reference | See figure 9 in Appendix B |

**3.2.7 Functional Requirement 2.2.7**

|  |  |
| --- | --- |
| Description | The system shall allow the user to screenshot the devices view, and save it to the devices local storage |
| Sequence of Operations | The user will select the screen shot UI button, resulting in the application screenshotting/saving the users view to the device |
| Exceptional Handling | Display an error message, and stop the saving process |
| Inputs | Selected UI button, and the users current view |
| Outputs | Image of the user’s current view in a .png file |
| Reference | See figure 9 in Appendix B |

**3.2.8 Functional Requirement 2.2.8**

|  |  |
| --- | --- |
| Description | The system shall allow the user to adjust the brightness of each light |
| Sequence of Operations | The user will select an already placed light, open the radial menu, select the adjust brightness menu option and change the brightness using the UI controls |
| Exceptional Handling | Display an error message and set the lights brightness to the default value |
| Inputs | The selected light, and the new brightness value |
| Outputs | Adjusted brightness of the selected |
| Reference | See figure 9 in Appendix B |

**3.2.9 Functional Requirement 2.2.9**

|  |  |
| --- | --- |
| Description | The system shall allow the user to change the color of each light |
| Sequence of Operations | The user will select an already placed light, open the radial menu, select the color picker menu option and change the lights color using the UI controls |
| Exceptional Handling | Display an error message and set the lights color to the default RGB value resulting in the color white |
| Inputs | The selected light, and the new color value |
| Outputs | Adjusted color of the selected |
| Reference | See figure 9 in Appendix B |

**3.2.10 Functional Requirement 2.2.10**

|  |  |
| --- | --- |
| Description | The system shall allow the user to change the range of each light |
| Sequence of Operations | The user will select an already placed light, open the radial menu, select the adjust light range menu option and change the lights range using the UI controls |
| Exceptional Handling | Display an error message and set the lights range to the default value |
| Inputs | The selected light, and the new range value |
| Outputs | Adjusted range of the selected |
| Reference | See figure 9 in Appendix B |

**3.2.11 Functional Requirement 2.2.11**

|  |  |
| --- | --- |
| Description | The system shall allow the user to change the lighting effects applied to all lights |
| Sequence of Operations | The user will select an already placed light, open the radial menu, select the adjust camera menu option and change the lighting effects applied to all lights |
| Exceptional Handling | Display an error message and revert the lighting effects to their default values |
| Inputs | The selected light, and the new effect values |
| Outputs | Adjusted effect intensity for all placed lights |
| Reference | See figure 9 in Appendix B |

**3.2.12 Functional Requirement 2.2.12**

|  |  |
| --- | --- |
| Description | The system shall allow the user to turn the simulated light on and off |
| Sequence of Operations | The user will select an already placed light, open the radial menu, and select the toggle light menu option |
| Exceptional Handling | Display error message, and toggle the selected light to the ON position |
| Inputs | Android UI toggle control |
| Outputs | Altered light intensity between zero and the maximal value |
| Reference | See figure 9 in Appendix B |

**3.2.13 Functional Requirement 2.2.13**

|  |  |
| --- | --- |
| Description | The system shall allow the user to delete a light fixture |
| Sequence of Operations | The user will select an already placed light, open the radial menu, and select the delete light menu option |
| Exceptional Handling | Display an error message and stop the deletion process |
| Inputs | The selected light |
| Outputs | The selected light is removed from the view |
| Reference | See figure 9 in Appendix B |

**3.2.14 Functional Requirement 2.2.14**

|  |  |
| --- | --- |
| Description | The system shall allow the user to move a light fixture |
| Sequence of Operations | The user will select an already placed light, and hold their finger over the light until the move command is initiated |
| Exceptional Handling | Stop the light from moving until initiated again |
| Inputs | The selected light |
| Outputs | The new light position |
| Reference | See figure 9 in Appendix B |

**3.2.15 Functional Requirement 2.2.15**

|  |  |
| --- | --- |
| Description | The system shall allow the user to rotate a light fixture |
| Sequence of Operations | The user will select an already placed light, open the radial menu, and select the rotate light menu option |
| Exceptional Handling | Reset the lights position/orientation, and stop the user from rotating the light |
| Inputs | The selected light |
| Outputs | The lights new orientation along the x, y, and z axis |
| Reference | See figure 9 in Appendix B |

**3.2.16 Functional Requirement 2.2.16**

|  |  |
| --- | --- |
| Description | The system shall allow the user to place and view multiple lights at the same time |
| Sequence of Operations | The user will place any number of lights they wish to see in the view |
| Exceptional Handling | Display and error message, and stop the user from placing more lights |
| Inputs | List of all placed and active lights since the start of the application |
| Outputs | NA |
| Reference | See figure 9 in Appendix B |

**3.2.17 Functional Requirement 2.2.17**

|  |  |
| --- | --- |
| Description | The system shall automatically keep track of the location of each placed light |
| Sequence of Operations | The user will place any number of lights they wish to see in the view, and the system will track their real-world positions until deleted by the user |
| Exceptional Handling | Stop the user from placing additional lights |
| Inputs | The users tap location on the screen |
| Outputs | The saved real-world position of the new light |
| Reference | See figure 9 in Appendix B |

3.3 Performance Requirements

The current section specifies the performance related requirements of the system. These requirements are supplied by the customer and have the potential to change through the early stages of development. As of revision 2.4, all performance requirements have been removed as they’re no longer relevant to the project.

3.4 Logical Database Requirements

The current section defines the logical requirements of the system, and databases. These include data entities, their relationships, constraints, accessibility, frequency of use, and retention over time. As of revision 2.4, all database requirements have been removed as they’re no longer relevant to the project

3.5 Design Constraints

At this current point in the design phase there are not too many design constraints to be considered. This section contains design constraints that can be imposed by any standards compliance and hardware limitations. This section also expands upon information given in section 2.4.

**3.5.1 Standard Development Tools:** The system shall be developed using the latest version of Unity and AR Foundation, as well as Android Studio version 3.1 or higher, with Android SDK Platform version 7.0 (API level 24 Nougat) or higher. Development will occur on API level 27 (i.e. Oreo version 8.1.0) using the default Android Emulator (version 27.2.9 or later) as part of Android studio (Google, N.D. a) (Google, N.D. b).

**3.5.2 Hardware RAM Limitation:** The system shall be expected to run smoothly on a device with 4 GBs or less of RAM.

# Appendix A:

The following section contains various images and diagrams referenced in the previous sections above; include the all UI mockups (figures 1 – 8)

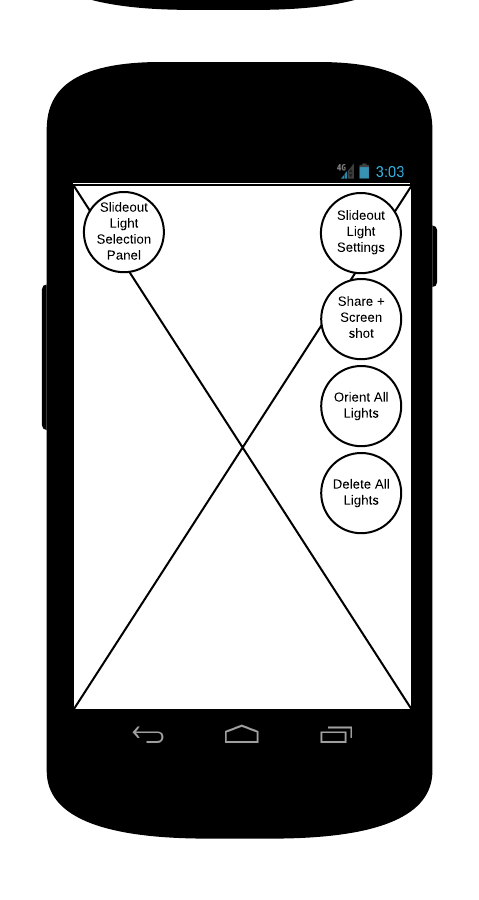


Figure 1. The base screen of the application. From here, users can see their surroundings using the camera module, and access the global options

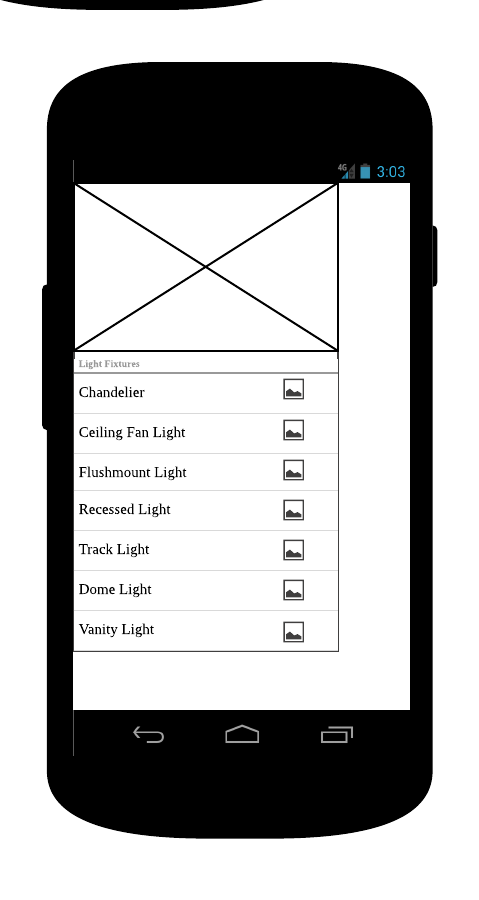


Figure 2. The slide out menu where users can access the different 3D light models from the provided list/table

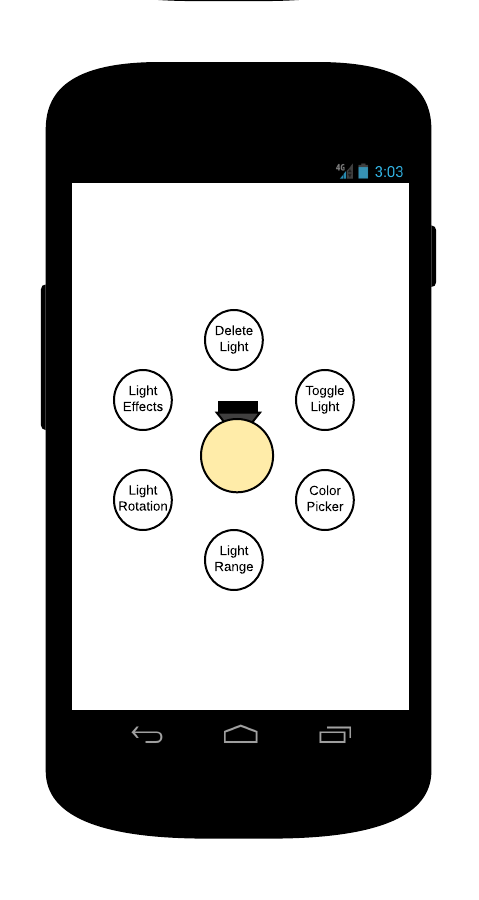


Figure 3. The light switch at the bottom left allows users to toggle the light intensity from zero (OFF) to max (ON)

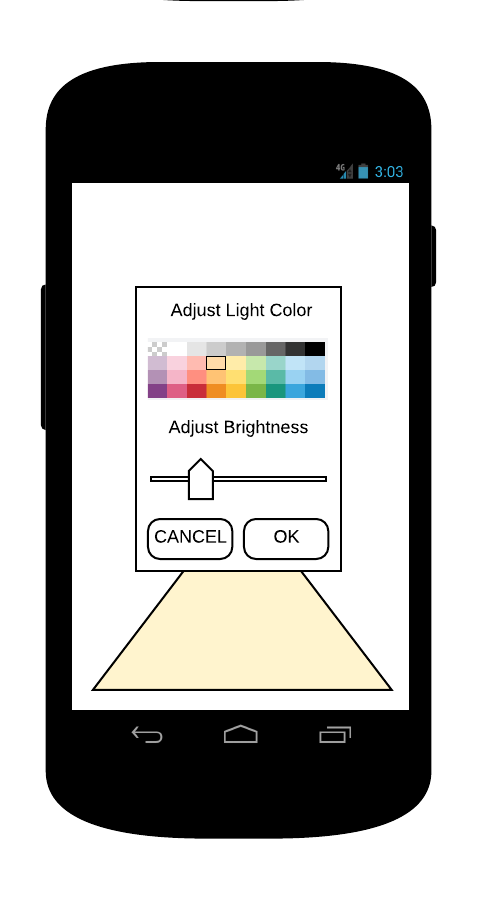


Figure 4. The light adjustment panel allows users to set the light intensity from zero (no light) to max. Setting the slider to the left will reduce the light levels

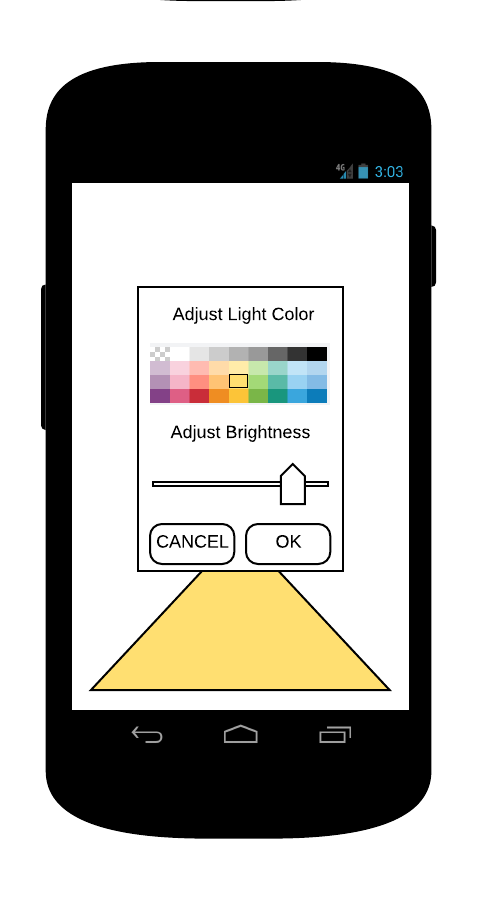


Figure 5. Adjusting the light intensity slider to the right, will increase the light levels

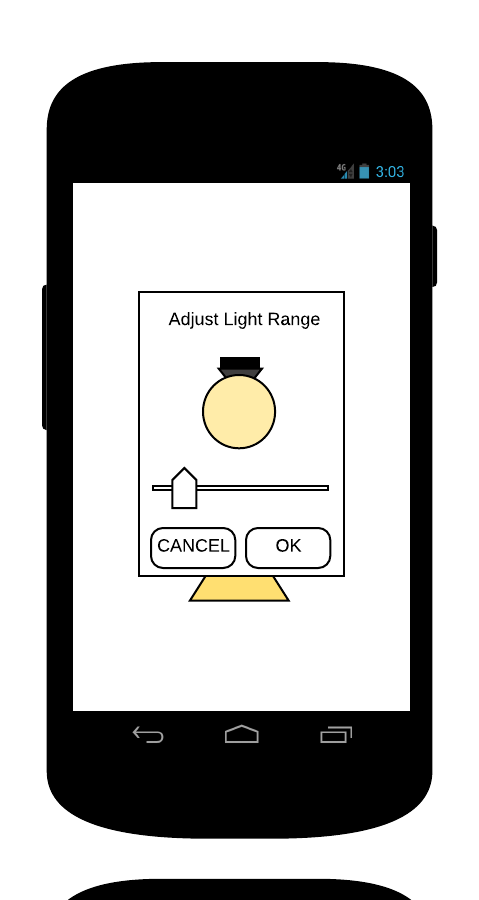
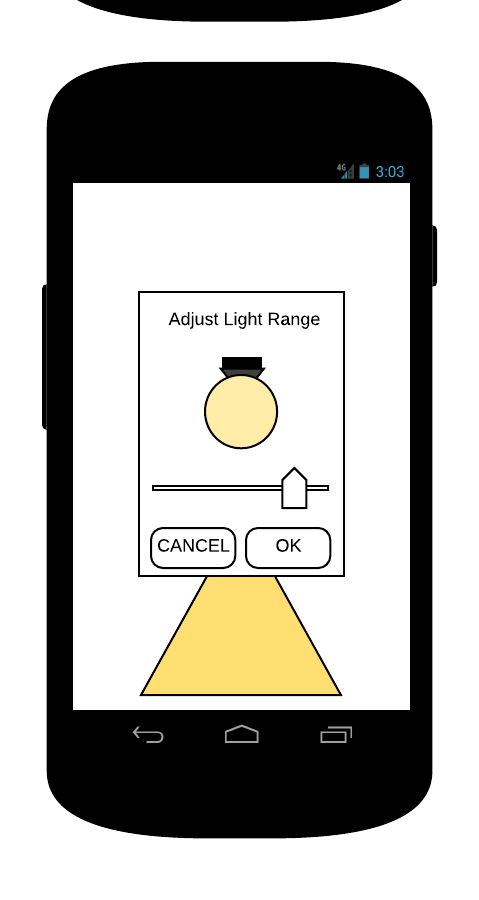
 

Figure 6. The light range adjustment window. Users can decrease the lights range by setting the slider to the left, and increase the range by sliding it to the right

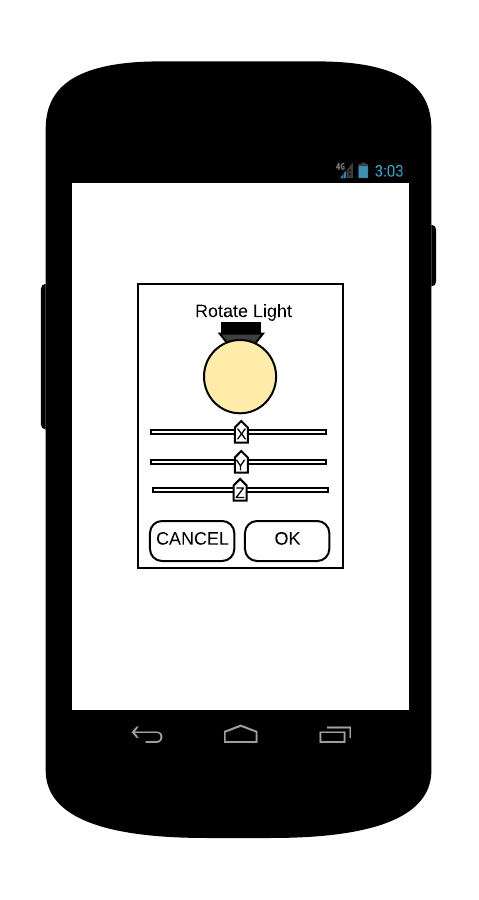


Figure 7. The light rotation window. Users can rotate the light along its X, Y, and Z axis by sliding the respective UI sliders to set the direction they wish to rotate the device

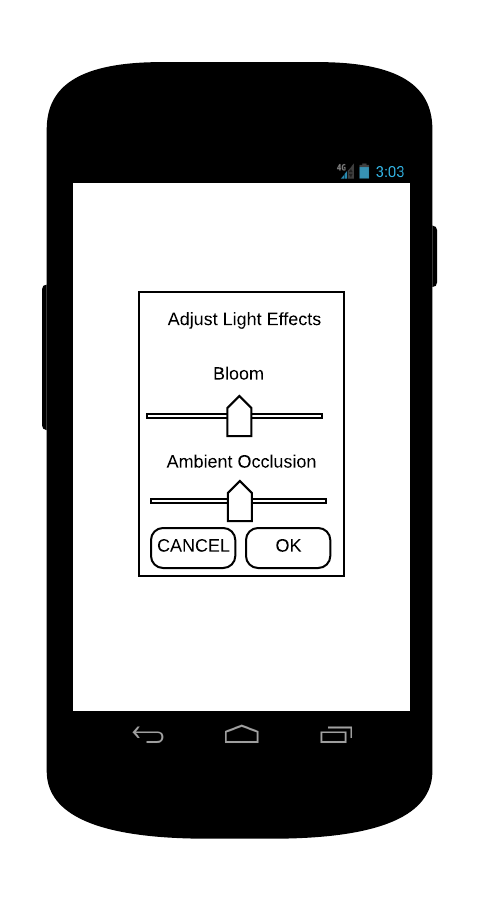
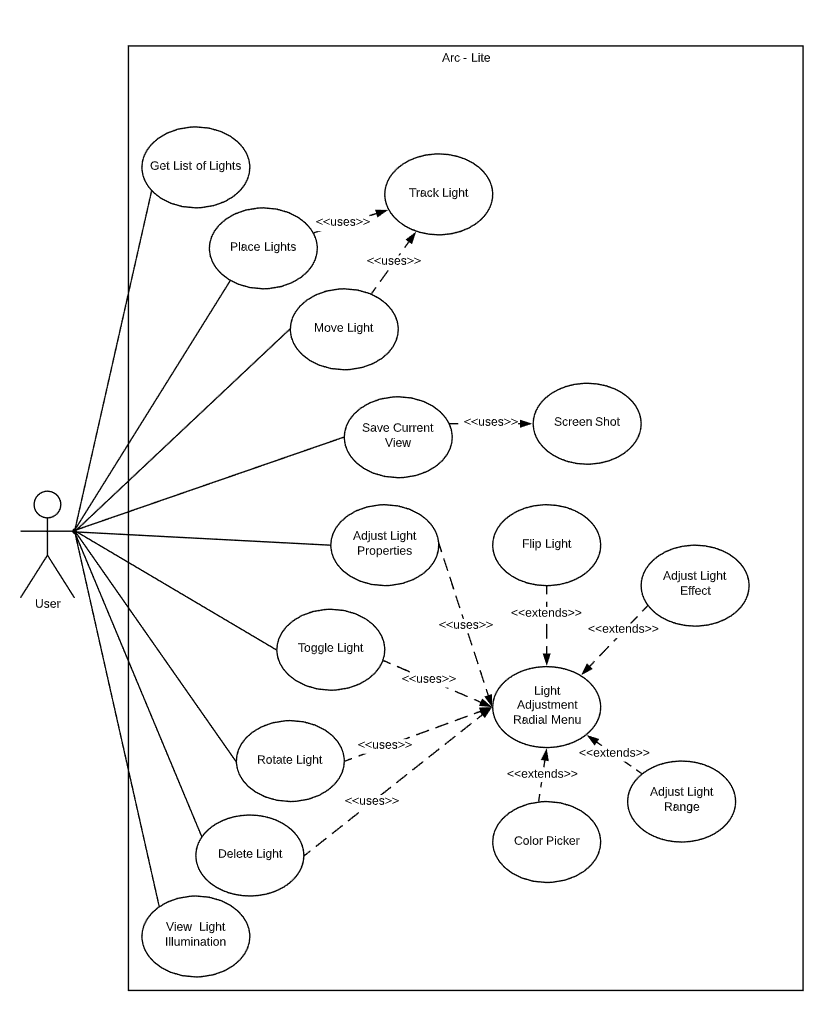


Figure 8. The light effects window. Users can set the intensity of various lighting effects affecting all the lights, including bloom, and ambient occlusion

# Appendix B:

The following section contains figure 9 which represents the use case diagram referenced in sections 2 and 3 above.



*Figure 9. The UC diagram shows an actor-oriented view of the system.*