**Software Requirements Specification**

**for**

**Project B.U.L.B**

**Version .05 approved**

**Prepared by Joshua Lazaro, Melissa Cheung , Gerardo Granados , Angelo Onato**

**Group 7 Company**

**October 09, 2017**

# **Table of Contents**

Table of Contents.................................................................................................................<pg 2>

Revision History...................................................................................................................<pg 3>

1. Introduction..............................................................................................................<pg 3>

1.1. Purpose........................................................................................................<pg 4>

1.2. Intended Audience and Reading Suggestions..............................................<pg 4>

1.3. Product Scope...............................................................................................<pg 4>

1.4. Definitions, Acronyms, and Abbreviations .................................................. <pg 5>

1.5. References....................................................................................................<pg 5>

2. Overall Description...................................................................................................<pg 5>

2.1. Product Perspective......................................................................................<pg 5>

2.2. Product Functions.........................................................................................<pg 6>

2.3. User Classes and Characteristics.................................................................<pg 6>

2.4. Operating Environment.................................................................................<pg 7>

2.5. Design and Implementation Constraints.......................................................<pg 7>

2.6. User Documentation.....................................................................................<pg 7>

2.7. Assumptions and Dependencies...................................................................<pg 7>

2.8. Apportioning of Requirements.......................................................................<pg 7>

3. External Interface Requirements..............................................................................<pg 8>

3.1. User Interfaces..............................................................................................<pg 8>

3.2. Hardware Interfaces......................................................................................<pg 8>

3.3. Software Interfaces.......................................................................................<pg 8>

3.4. Communications Interfaces...........................................................................<pg 9>

4. Requirements Specification......................................................................................<pg 9>

4.1. Functional Requirements............................................................................<pg 10>

4.2. External Interface Requirements.................................................................<pg 11>

4.3. Logical Database Requirements.................................................................<pg 12>

4.4. Design Constraints......................................................................................<pg 13>

5. Other Nonfunctional Requirements........................................................................<pg 13>

5.1. Performance Requirements........................................................................<pg 13>

5.2. Safety Requirements...................................................................................<pg 14>

5.3. Security Requirements................................................................................<pg 14>

5.4. Software Quality Attributes..........................................................................<pg 14>

5.5. Business Rules...........................................................................................<pg 15>

6. Other Requirements...............................................................................................<pg 15>

Appendix A: Glossary.........................................................................................................<pg 15>

Appendix B: Analysis Models.............................................................................................<pg 15>

Appendix C: To Be Determined List...................................................................................<pg 15>

# 

# 

# **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# **1. Introduction**

Project Bulb will bring ease into operating everyday electronic devices. This Document will break down in depth what exactly our application has to offer. What our application does is allow the user to turn on any type of electronic device(That is compatible with our app) using their very own mobile device. It also includes :

* Ability to change a LED color
* Ability to turn on anything inside your home(Anything compatible with our app)
* Making turning on a light from your phone faster than you can say fuck

The Introduction section of the Software Requirements Specifications (SRS) document should give an overview of the entire document. As you write this document, remember that the purpose of this document is to tell software engineers WHAT the software is supposed to do. The details on HOW the software will be implemented should be specified in the Software Design Document (SDD).

## **1.1 Purpose**

Purpose of this document is to explain in depth on what exactly Project Bulb does. Going into spec

Identify the purpose of this document and the product whose software requirements are specified in this document. Be sure to mention the version / revision / release number of the software and also mention if the document covers all aspects of the software or only a small part of the software. Do not iterate over the table of contents here. Give a general overview of what this document contains.

## **1.2 Intended Audience and Reading Suggestions**

This section should mention what types of readers the document is intended for such as developers, project managers, marketing staff, users, testers, and documentation writers.

Describe what this SRS contains and how it is organized. If each type of reader should interpret this document differently, give a suggested reading sequence for the document that is relevant to that particular reader type. Example: A tester might only be interested in the detailed requirements list, or the marketing staff may only be interested in the very high level descriptions of the project.

## **1.3 Product Scope**

In this section:

● Identify the software product(s) the be produced by name.

● Explain what the software will do. If necessary mention what the software will not do.

● Describe how the software will be used once released, include benefits of the software, objectives, goals, etc.

This section should give a high-level summary of the software. Do not list all of the requirements here.

* **We are making a mobile app to control a LED lights also with all the other function.**
* **Such as changing color, adding alarm, timer.**
* **Adjust brightness, with other different mode**
* **Mobile app can connect to the light by using a bluetooth device. With make it fast and easy**

## **1.4 Definitions, Acronyms, and Abbreviations**

Provide definitions for any relevant terms, acronyms, and abbreviations that are necessary to understand the SRS document. This information may be listed here, in an appendix at the end of the document, or in a completely separate document. If the information is not directly listed in this section provide a note that specifies where the information can be found.

* B.U.L.B. - Beautifully Used Light Bulb

## **1.5 References**

This section should provide the following:

● A complete list of all documents referenced in the SRS.

● A complete list of any Web addresses referenced in the SRS.

● For each reference mention the title, author, version number, date, and the source or location of the reference.

● Generally you should provide enough information for each reference so the reader of the SRS can easily obtain copies of these references.

References may include items such as: user interface style guides, contracts, standards, other SRS documents, use case documents, supplemental documents that were researched while writing this SRS or while designing the software.

# **2. Overall Description**

This section explains the general facts that affect the software and its requirements. You should not state any specific requirements in this section. Here you will provide a background for the actual list of requirements that will be defined in section 4. The point in this section, is to give a high level description of the requirements in plain English for the benefit of the customer / client.

## **2.1 Product Perspective**

This section should show how the software may relate to other products. If the software is completely independent of any other products / systems, you should also mention that here.

If the SRS describes software that is a component of a larger system, this section should describe how the requirements of this software relate to the larger system. Be sure to mention and identify interfaces between the larger system and the software that this SRS defines.

This section should also mention how the software in this SRS compares to similar systems currently on the market (if any exist). Mention similarities / differences / motivations for creating this software when compared to an existing product. If the software is a completely new innovation, be sure to mention that as well.

You may provide a block diagram here showing the major components of the larger system, external interfaces, and how your software fits in with the larger system. This is NOT a design or architecture picture. It is simply here to provide a context for the reader to understand where your software fits with the larger system. Your software should be shown as a black box. Remember, it is the purpose of the Software Design Document to present the internal workings of your product.

* This product is dependent on a bluetooth module in order to connect to the light.
* Similar products in the market include features such as…
  + Changing brightness, mode, and color
  + Linking app to alarm
* The motivation for this project is to allow users to control bluetooth lights with ease as well as giving personalization through the function of changing color of the lights and choosing different modes for the user’s style.

## **2.2 Product Functions**

The product functions consist of having an ability of controlling lights at your home from your phone.

* Turn on Anything that is connected to the app.
* Linking the lights to an alarm.
* Changing the color of the light
* Selecting different modes for certain situations like reading mode.
* Allow lights to go with the beat of the music
* Control multiple lights through different channels

This section provides a summary of the major functions that the software will perform. The functions should be organized in a way that makes them understandable to the customer / client or anyone who is reading this document.

You can provide a bullet list summary of each function and / or provide simple diagrams to show the different functions and their relationships. Again, these diagrams do not show the design of the product, just the relationships between the functions.

## **2.3 User Classes and Characteristics**

* Young adult
* Teenage
* Restaurant design
* Persons with difficulties walking

## **2.4 Operating Environment**

The BULB is being developed to function in an android environment. It is compatible with android version 7.0 -- Nougat. Planned release for iOS.

## **2.5 Design and Implementation Constraints**

1. Model of android device
2. Permissions granted to app
3. Linked Relationships with other apps
4. Android Version Compatibility

## **2.6 User Documentation**

List any user documentation (user manuals, online guides, tutorials, etc) that will be delivered along with the software. Specify in what format each document will given.

* User manuals in small booklets that give out instruction how to use the product.
  + User manuals will also be available online as well.
  + User manuals will also contain safety warnings

## **2.7 Assumptions and Dependencies**

* Third party component such as bluetooth.

## **2.8 Apportioning of Requirements**

* iOS Device
* iOS Version X.X

List any requirements that might be delayed until future versions of the system.

# **3. External Interface Requirements**

Again, this section is a high level description for the non-technical people who may be reading this document. Everything in this section is to detail how your software interacts with any external interfaces, whether these are other software or even hardware interfaces.

## **3.1 User Interfaces**

Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.

Be sure to mention any requirements that must conform to the Americans with Disabilities Act.

## **3.2 Hardware Interfaces**

Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.

This is not a description of hardware requirements such as "This program will run on a PC with 16 gigabytes of RAM." This section details the actual hardware devices that your application will interact with. Items such as robotics components, lighting, audio / video equipment, etc. If your software does not have an hardware interface requirements, then state this instead.

* We using android as our main software to connect to the bluetooth(hardware).

## **3.3 Software Interfaces**

List any other software products and interfaces that your requirements must utilize. For each required product, list the name, version number, and source.

Document any APIs that your software will have to access in order to interact with other software products. Also document how your software will communicate / pass information to the external software.

Example, your customer uses MySQL version 1 and you are required to use that in your design, then you must specify this here.

This section is NOT for specifying software that you think would be good to use. This is for customer-specific requirements that you HAVE to interact with.

* Since we using the android app to connect with other function on phone such as alarm and music player. Because now a day a lot of people want to access all the stuff just on the phone by one click. So we decide to use android app.

## **3.4 Communications Interfaces**

Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.

# **4. Requirements Specification**

This section contains all the necessary software requirements with enough detail to allow designers to accurately design the software to satisfy those requirements, and to allow testers of the software to verify that all requirements have been satisfied. The requirements should include a description of every input to the system, every output, and all functions performed by the system in response to an input or output.

The biggest thing to remember is that this section is for the software developers (technical people) while the previous sections were for the customers / non-technical people.

Also remember that this is not HOW things will be implemented, but WHAT will be implemented.

Requirements should be written according to the following:

1. Specific requirements should be correct, unambiguous, complete, consistent, ranked for importance and / or stability, verifiable, modifiable, and traceable.

2. Specific requirements should be cross-referenced to earlier documents that are relevant.

3. All requirements should be uniquely identifiable using a consistent numbering system, i.e. 1.1, 1.2, 1.1.2, and so on.

4. Requirements should be organized in a logical manner to provide the most readability.

Use the following format for each requirement:

The system shall... (this means this requirement is mandatory).

The system should... (this means a desired feature, but may be delayed until later).

This system may... (An optional, nice-to-have feature that might not be implemented).

Remember to number each requirement for traceability. Use a system such as 1.1, 1.1.1, 1.1.2.1, and so on. Each requirement need to be testable. Avoid statements that are general and vague such as "The system shall be easy to use." or "The system shall be developed using good software engineering practices."

Do not include examples. Remember that this is a specification and the designer should be able to read this and build the system without having to bother the customer again. Every minute detail must be documented here.

EVERYTHING in section 4 must be written following the above guidelines.

## **4.1 Functional Requirements**

1. The system shall…
   1. Allow users to connect to LED lights via bluetooth connection.
      1. Allow users to switch the connected lights ON and OFF
2. This system should…
   1. Allow various controls over the conditions of the LED
      1. Have a timer feature
         1. Allow users to set active hours for LED
         2. Set LED to turn off after a designated amount of time
      2. Allow for different built in light modes
         1. Study mode
         2. Reading mode
         3. Night Light mode
      3. Allow users change the color of the LED light
      4. Allow users to adjust brightness
         1. Ability to save the brightness setting as a custom light mode
3. This system may…
   1. Links to other apps in the phone (with permission) for additional functions
      1. Allow users to link lights to their music player
         1. LED will blink with the beat of the music being currently played
      2. Allow users to link to their alarm app on their phone
         1. User is able to set the LED to automatically turn ON when the alarm sounds off

## **4.2 External Interface Requirements**

Name of item

* B.U.L.B

Description of purpose

* Allow user to adjust the LED light in their own way.

Source of input or destination of output

* We are using mobile app as an input to connect with the bluetooth and then use that as an output to connect to the LED light.

Valid range, accuracy and/or tolerance

* Depends on the strength on the connection

Units of measure

* Approximately 50 meters from your house

Timing

* 0.5 seconds to turn off light

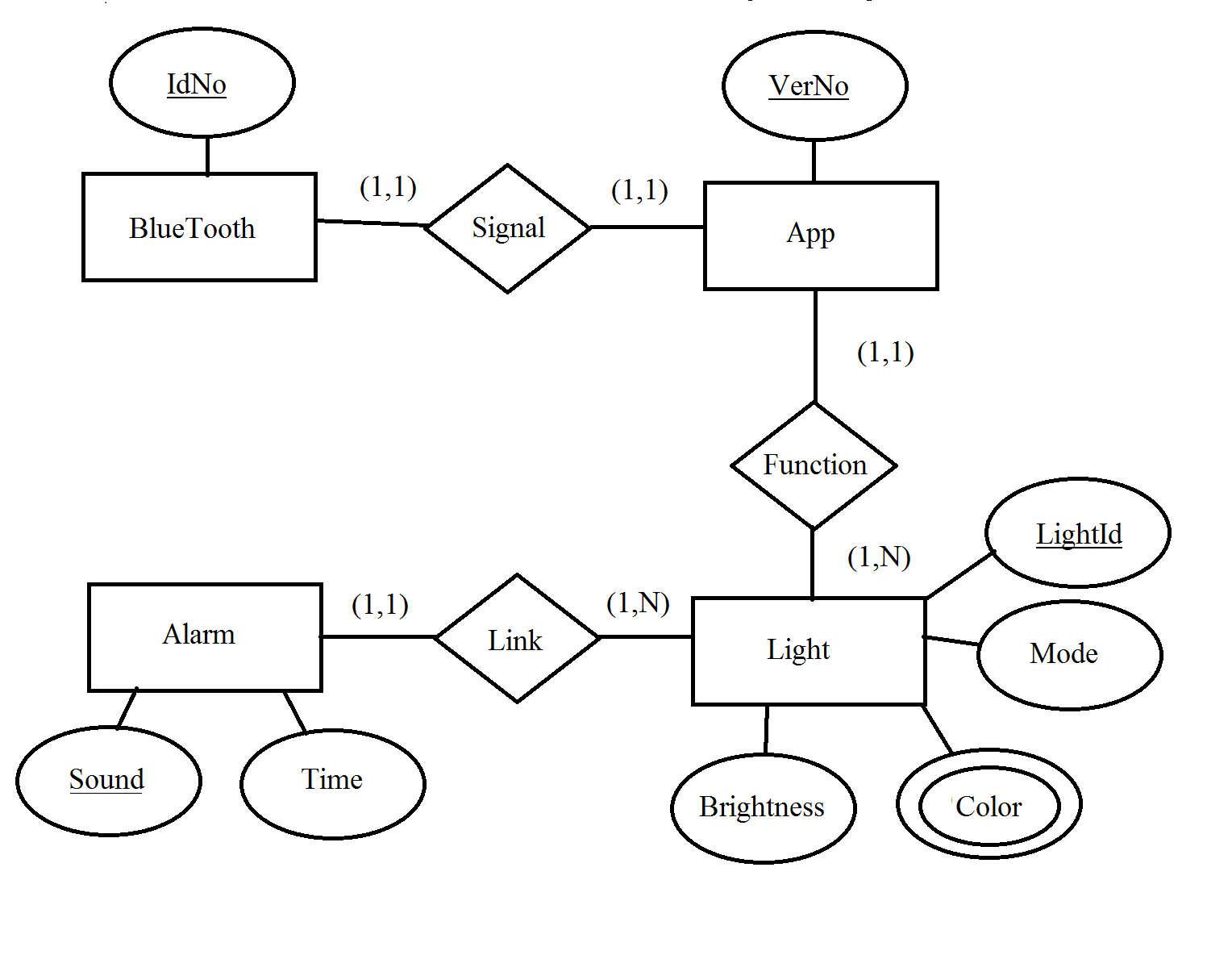
Relationships to other inputs/outputs

* Mobile app is for control the LED light(input), then it will send a signal to the bluetooth device. The bluetooth will connect the LED light, then we can easily to control the light whatever we want by using the app.

Screen formats/organization

* The screen size can be adjust by the user

**4.3 Logical Database Requirements**



## 

## **4.4 Design Constraints**

1. Model of android device
   1. There are various android devices so some models may not work as well as others
2. Permissions granted to app
   1. Some functions of the app may not function without specific permissions granted
3. Linked Relationships with other apps
   1. Multiple music players may cause issues
   2. Alarm app on different model phones may function differently
4. Android Version Compatibility
   1. Unknown interactions with older or newer Android Versions

# **5. Other Nonfunctional Requirements**

## **5.1 Performance Requirements**

This section specifies any numerical / statistical requirements imposed on the software such as:

● The number of terminals to be supported

● The number of simultaneous users to be supported

● Amount and type of information to be handled

Dynamic numerical requirements may include, for example, the numbers of transactions and tasks and the amount of data to be processed within certain time periods for both normal and peak workload conditions.

All of these requirements should be stated in measurable terms. For example, "95% of the transactions shall be processed in less than 1 second" rather than, "An operator shall not have to wait for the transaction to complete."

(Note: Numerical limits applied to one specific functional requirement are normally specified as part of that requirement and should be listed in section 4. This part is more for performance / statistical requirements imposed on the entire system as a whole.)

## **5.2 Safety Requirements**

Caution: To those who are prone to having seizures due to flashing lights, bold, regular patterns, or regular moving patterns.

Taking into consideration that you keep a good watch for anything that might cause a fire hazard.

Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.

## **5.3 Security Requirements**

Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.

* Since this product is using a bluetooth device to connect with the phone app, we will need to use password or to access it.
* Bluetooth device will have a code to connect with the app, not anyone can access it

## **5.4 Software Quality Attributes**

Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.

* Easy to use
* Highly customizable

## **5.5 Business Rules**

List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.

* The creators of this product will have exclusive use of coding. Certain circumstances for a creator to code in the product is to update, fix, or delete.
  + Consumers cannot have access to this feature as they may modify code to cause problems

# **6. Other Requirements**

Define any other requirements not covered elsewhere in the SRS. This might include internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project in this section.

This is pretty much a catch-all for things that do not fit in a previous section.

# **Appendix A: Glossary**

# **Appendix B: Analysis Models**

Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.

# **Appendix C: To Be Determined List**

Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.

**General Template Directions:**

● This page should not appear in the final document.

● Replace any **<Insert x here...>** placeholders with the actual text. Yes this also means replacing the angle <> brackets as well. Your document should not have ANY **blue** text in it. If the text had any kind of formatting (bold, italics, font size) keep the same formatting. DO change the font color to black.

● Remove any **<green text>** placeholders. These are descriptions about what to write in each section and may also provide examples of what to write. Do not simply copy / adapt the wording to fit your project. All green text should be replaced with your own text or removed completely in some cases.

● **DO NOT** change the formatting of the document this includes:

• font sizes

• font weights

• italics NOT in angle brackets

• margins

• spacing

• adding or removing sections of the document.

● Each major numbered section (1, 2, 3 etc.) should start on a new page.

● When the document is finished, update the table of contents with the correct page numbers, and make sure the numbers are properly aligned to the right.