Fall 2019 / CSC-450

Feasibility Report

1. Client Information

Our client is (cascading down):

United States Department of Energy (DOE)

Office of Energy Efficiency and Renewable Energy (EERE)

Building Energy Research & Development

Building Technology Office

Advanced Lighting Program

For succinctness, the client is referred to as “the client” or “DOE” in the rest of this document.

DOE Contacts:

Dr. Brian Walker

Solid State Lighting Technology Manager

EERE Building Technology Office

202-586-5000

brian.walker@ee.doe.gov

Dr. Wyatt Miller

AAAS Science & Technology Policy Fellow

EERE Building Technology Office

wyatt.miller@ee.doe.gov

Morgan Pattison

Senior Technical Advisor

U.S. DOE Lighting R&D Program

(awaiting contact info from Brian)

2. Team Member Information

Backend / Database Team:

Lily Donaldson (client liaison) - ld9300a@student.american.edu

Yuhao Li - yl4139a@student.american.edu

Zhiqi Lu - zl3553a@student.american.edu

Front End Team:

Elisabetta Gabriele - eg3174a@student.american.edu

Chuyan Cheng - cc7997a@student.american.edu

Rebecca Sakaguchi - rs9068a@student.american.edu

Floating:

Anthony Baron - ab1322a@student.american.edu

---

3. Project Overview

Project Name: Illuminant or SpectraSearch, pending client feedback

I. Explanation of Industry Concepts

Lighting products and lighting in general can be characterized by two lighting properties:

1. Spectral power distribution
2. Optical distribution

Spectral power distribution (SPD) is the distribution of optical power (y) over different wavelengths of light (x). Wavelengths of visible light fall between 380-400 and 700-740 nanometers on the electromagnetic spectrum, from purple to red. UV wavelengths range from about 10 - 400 nanometers. Infrared (IR) wavelengths range from about 700 nanometers to 1 millimeter. Visible light is particularly important for human lighting, and UV and IR rays may be helpful for horticultural lighting, among others.

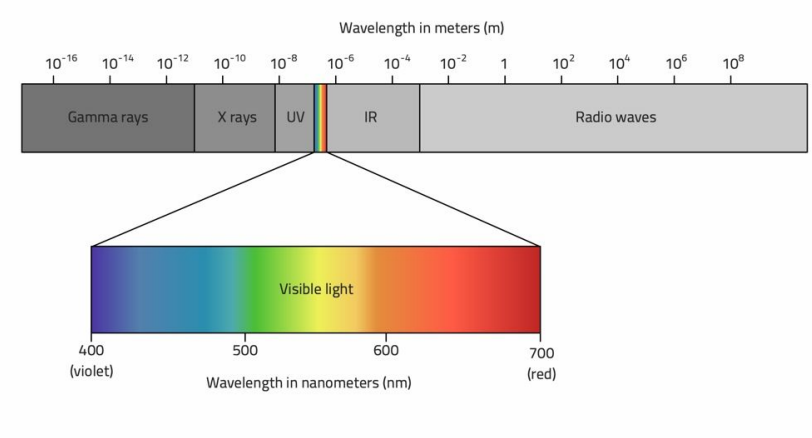

Figure 1: Visible Light on the Electromagnetic Spectrum[1]

SPD graphs plot these wavelengths in nanometers against their respective optical powers normalized to percentages of a whole (0 to 1).

Optical distribution is the angular distribution of light. It is categorized into five different types. Type one has two main concentrations of light going in opposite directions. Type two is meant for more focused areas or narrow roadways. In the third type of distribution, light projects outward and fills even larger areas or roadways. Type four is meant to be mounted on the side of a parking lot and projects in a semi-circular path. Finally, type five optical distribution produces light with the same amount of power in all directions; it is typically used in large spaces like commercial parking lots.[2]

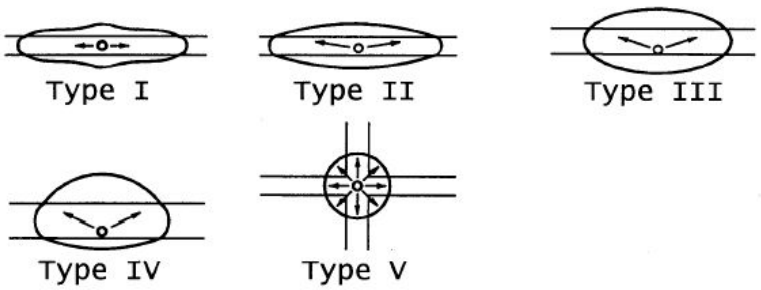

Figure 2: Optical Distribution Types[3]

From these two aspects of lighting performance (SPD and Optical Distribution), all other lighting properties in a given space or plane in space can be calculated.

[1] https://www.radio2space.com/components-of-electromagnetic-spectrum/
[2] https://eyelighting.com/lighting-technology-education/led-lighting-basics/led-distribution-types
[3] http://www.maxliteuniversity.com/common-lighting-terms-definitions/

In particular, lumens, the photometric output of a light source can be calculated from the SPD. Other new metrics of interest can also be calculated such as photosynthetically active radiation (PAR), which is the light radiation that plants absorb; or melanopic lumens, which describe the non-visual impact of light mediated through the eye that impact hormone secretion, particularly melatonin.

Historically, the lighting industry has primarily used the lumen to characterize light but with LEDs, new applications for horticulture and physiological responses can be considered, and these metrics can be calculated from the SPD. New color qualities can also be calculated from the SPD.

II. The Database

The client's main request is a publicly available database of lighting instruments including SPD and optical distribution data for each instrument. The database should also allow for storing response spectra.

III. The Interface

The client's secondary request is an evaluation tool connected to the database of lighting instruments. The evaluation tool will consist of a main search for lighting instruments and corresponding interactive data (SDP graphs, manufacturer information), which will allow professionals to evaluate lighting products against current and future requirements.

IV. The Why are We Doing This

This product will **help lighting professionals evaluate lighting products** against current and future requirements. The SPD and optical distribution data will be used as inputs for advanced lighting design modeling to provide **improved lighting delivery and optimized color qualities** for the lighting application.

The features of this software are not currently available to the professional lighting community and initial client and general user interviews suggest these features will be beneficial to the community, especially as an open source tool.

V. The Client / Intended Users

There are three intended user groups for this product: Industry Professionals, Academics, and Novices. Industry Professionals work in the lighting industry as architects, engineers, lighting designers, etc. Academics are researchers in lighting, optics, architecture, perception or psychology, etc. We will consider a novice user to be generally familiar with the lighting industry, but less apt than Industry Professionals or Academics (i.e. students).

User Group Testing Subjects:
Industrial: Department of Energy (DOE)
Academic: Dr. Arthur Shapiro (American University Professor) and/or DOE researchers
Novice: Lily Donaldson (lighting student)

---

4. Preliminary Requirements

The software product is meant to allow users to search for lighting instruments and have access to relevant data associated with the instrument, so that lighting research for professionals and academics can become more efficient and accurate.

The product will implement an SQL database with organized data connected to a front-end user interface which allows the user to easily access lighting information. Database information will also be used to allow the front end of the application to compute further information for a lighting instrument. For instance, spectral power distribution data will be convolved with response spectra to output data like lumens, photosynthetically active radiation, and melanopic lumens.

We are assuming the intended user groups (Industrial, Academic, Novice) are, to an extent, expert users. All three user groups, including novices, will be familiar with the data types included and will be scientists or technicians in one form or the other. Because of this, we will focus our initial development on functionality rather than usability as this will be more valued by our intended user groups. For example, we believe that intended user groups (the client(s)) would value the ability to convolve lighting instrument data with response spectra over a ‘prettier’ or even easier to use interface.

We are still awaiting information on DOE servers/hosting, but the finished app/database will likely be held on DOE servers. We have not been told if it will be a separate web app from the DOE website (energy.gov) or if it will be a module on an existing website page. In any case, the client specifically requested a SQL database and we are sure that the front end coding (HTML/CSS, Javascript) will be accessible to the DOE website.

Functional Requirements (in order of importance)

- The database needs to be set up to store SPD and response spectra with appropriate data resolution and optical distribution information
- The interface needs to allow access to the database
- The interface needs to allow uploading lighting instrument data and response spectra data in a variety of formats
- The interface should allow users to grab response spectra data and instrument data from the database, convolve the two, and show the output SPD
- The interface should allow users to compare data on instruments

Non-Functional Requirements

- The interface should be clean and concise as not to fuzzy the data viewing
- The interface should display the instrument data and response data in a consistent, clean, and concise manner
- Convolution time should be minimized

---

5. Deliverables

- Report 2 (Progress Report) and Survey 2
  Due 10/11/19
- **Sprint 1: Database of Lighting Instruments**
  Deliverable(s): Database schema and SQL Database
  Due 10/14/19
- **Sprint 2: Preliminary User Interface**
  Deliverable(s): Functional user interface (though not connected to database)
  Due 10/14/19
- **Sprint 3: Connect ‘em Up**
  Deliverable: Connect the database and user interface
  Due 10/28/19
- Report 3 (Progress Report) and Survey 3
  Due 11/1/19
- **Sprint 4: Interface Interactivity**
  Deliverable: (1) Allow users to compare data on different instruments. (2) Allow users to upload Lighting Instrument data and response spectra into the database in the following formats:

- Spectrometer measurements (may have different resolution)
- Standardized IES file formats
- Images with spectra

Due 11/4/19

- **Sprint 5: Response Spectra Function**
  Deliverable: Software functionality to convolve LI SPD with response SPD (i.e. lumens, photosynthetically active radiation (PAR), melanopic lumens)
  Due 11/25/19
- **Integration with client servers / website and Fully packaged software product**
  Due 12/11/19

---

6. Process

The team will use an Agile approach to the project. The team will develop elements of the project incrementally, analyze progress, and address changes during standups/scrum meetings and larger sprint meetings. Standups or scrum meetings will occur every Monday starting from 9/9/2019 to 12/11/2019. All team members are required to attend and the meeting will address what members have been working on, what they plan to work on next, and any questions they have. There will be 3 sprint meetings in which members of the team will showcase their progress on elements they are developing. All team members will comment and make suggestions on members' progress. If changes need to be made, those team members will go back and address them within their respective sub-teams. Communication will be handled on a daily basis through Slack and Messenger.

We will regularly push our code on our respective git branches. We will not merge our branches back into the master branch without testing and full team approval. We will keep our kanban board up to date and pull the 'Andon cord' if we discover a bottleneck.

We will regularly involve the customer through weekly meetings and intermittent prototypes to receive feedback. We will provide the customer with specific details for project requirements. Further information on client meetings is addressed below in the visibility plan.

---

7. Outline of Plan

Pre-Sprint

Deliverable(s): Feasibility Study

Report 1 (Feasibility Report) and Survey 1 (9/13/19)

- Preliminary design meeting with client *(08/29/19)*
- Client software proposal *(09/04/19)*
- Pre-Sprint Scrum 1 *(9/9/19)*

  What we did: Discussed general idea for project, Went over client's proposal, Divided into teams for Sprint 1 (Database) and Sprint 2 (Front-end web app) which will run concurrently, Discussed what next Sprints could be, Learned the basics of lighting measurement (SPD and CIE), Discussed potential backend languages/frameworks (PHP, Java, or Python) and potential UI/UX testing, Discussed Risk Analysis and feasibility study

  What we will do before next meeting: Read up on client's industry, Write feasibility study (Google Drive), Join Github organization, Clarify data structure and some parts of proposal with client, Discuss hosting with client/client's IT department

  Things we/someone need(s) help on: Clarification of several things with client
- Client meeting scheduling and proposal questions *(awaiting)*
- Pre-Sprint Scrum 2 *(9/16/19)*

  Post-feasibility study; Begin Sprints 1 and 2

Sprints 1 and 2 (9/16/19 - 10/14/19)

Sprint 1: Database of Lighting Instruments

Deliverable(s): Database schema and SQL Database

- Data Struct Design
  - Industry research
  - Confirm struct items with client
  - Discuss data types with client
  - Finalize Data Struct Design
- Database Design
  - Database Schema
  - List potential database methods
- Database implementation with sample data items
  - Create database (SQL)
- Addition of client data

Sprint 2: Preliminary User Interface (concurrent with Sprint 1)

Deliverable(s): Functional user interface (though not connected to database)

- Industry research on lighting instruments
- Wireframes (blueprints of UI design, loose idea of where elements will live on the page)
- Main pages set up with react js/css
- Homepage (search bar), searching ‘data’
- Main search for lighting instruments with suggestions for names of lighting instruments as a user types
- Hovering over an option with produce brief pop-up overview of the main elements associated with that lighting instrument.
- Clicking on the instrument brings the user to a page with all the information and data relevant to instrument.
- Modals for if a user clicks on a piece of information within that lighting instrument data.
- Create an outline of page that has space for SDP graphs, manufacturer information, name of lighting, optical distribution information, and commercial information.
- Showed the search history for reference.

- Scrum *(9/30/19)*
- Scrum *(10/7/19)*
- Report 2 (Progress Report) and Survey 2 *(10/11/19)*

Sprint 3: Connect ‘em Up (10/14/19-10/28/19)

Deliverable: Connect the database and user interface. Design specifics are added to interface.

- User interface design
    - Creation of personas + scenarios (imagining different people that would use the software, and their top goals and priorities)
    - Map out user flows (how users will take actions through the system, separate from user stories)

- Backend

    - Finishing the basic structure for the back end and format to be easy to add new features in the future
    - Getting the first version of the functioning product
    - Making sure ports of the database and UI are in good format and easy to call. Since it is the first time to connect the database and front - end and get an initial basic version of a complete product. If we can default the format at the early stage of the project, it will reduce much unnecessary workload for the rest of the project.
    - Giving the feedback to both the front - end and database group.
- Scrum *(10/14/19)*
- Scrum *(10/21/19)*

Sprint 4: Interface Interactivity (10/28/19 - 11/4/19)

Deliverable: Allow users to upload Lighting Instrument data into the database and compare instruments.

- Section of UI for allowing users to add new lighting instrument categories, and then data within that category
- Users can also select a category of instrument from a dropdown and add another instrument to that category, or pick an existing instrument and add relevant data.
- For comparisons of lighting instruments, there can be a comparison cart, and users can drag different lighting instruments from a dropdown of search suggestions into a comparison cart. When they have all their desired comparisons in the cart, they can click on the cart, which will reveal x number of instrument modals side by side (the team will consider the most efficient method of comparisons which requires the least amount of steps).
- Implement visual design of UI (expand beyond basic layout, introduce color, typographic choices, button styles, etc)

- Scrum *(10/28/19)*
- Report 3 (Progress Report) and Survey 3 *(11/1/19)*
- Scrum *(11/4/19)*

Sprint 5: Response Spectra Function (11/4/19 - 11/25/19)

Deliverable: Software functionality to convolve LI SPD with response SPD (i.e. lumens, photosynthetically active radiation (PAR), melanopic lumens).

- Use data and coordinates from database/json to set up SDP graphs.
- Perform the math to convolve the emitted SPD with the response SPD and provide resulting values in a table format
- Decide if we want to calculate this supplemental data prior to user loading page, or if there should be interactive buttons implemented to calculate on the spot.

- Scrum *(11/11/19)*
- Scrum *(11/18/19)*

Client Integration and Leeway / Additional Feature Time

- Integrate software with client servers / website
- Scrum *(11/25/19)*
- Scrum *(12/2/19)*
- Final Presentations, Report 4 (Final Report), Code Delivery, and Survey 4 *(12/11/19)*

---

8. Visibility Plan

- How will you keep in contact with the client and report progress? How often, and how, will you meet?
    - Lily (client liaison) will meet with the client regarding the software product at least once a week in person at the DOE Buildings Building and will communicate to the client as needed over email.
- How will you communicate among your team? How often, and how, will you meet?
    - We will have weekly scrum meetings on Mondays 10-11 AM in DMTI 202.
    - We will keep track of tasks with **Trello**.
    - We will communicate as needed via **Slack** or **Messenger**.
    - We will push our code to our sub-project **git** branches often and will keep documentation of our code up to date in the **Google Drive** so that our work is transparent to our team members.
    - We will keep our digital kanban board (Trello) up to date.

- We will separate our codes in stable and testing versions. When each team is trying to update new features, they will only update them to the testing branch to make sure the other team can do their own test without getting affected.

---

9. Business Considerations

Based on the client's requirements and legal concerns, the software product team takes the following business considerations into account:

(1) This product will be considered as an open-source software that can be freely used, modified, and shared;

(2) Once delivered, our team will be hands off regarding warranty for this product and will not be liable for its maintenance, security, distribution and performance

(3) (Restrictive) GNU General Public License, version 3 (GPLv3):

(a) All source code must be public;

(b) Modifications of the software must be released under the same license;

(c) Changes to the source code must be documented;

(d) This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. For a copy of the GNU General Public Licenses see <http://www.gnu.org/licenses/>

---

10. Risk Analysis

A.

a. Issue: Government Shutdown

Our client, the Department of Energy, is one of the branches of the government that is furloughed when there is a government shutdown. Due to this, we would likely not have contact with the client and, if we were in the client integration phase, would not be able to implement or test our application on their servers for the duration of the government shutdown. If they are to occur, government shutdowns are likely to occur between September and January.

b. Fallback plan: If a government shutdown is impending, we will try to get all of the information we need from the client to continue developing in their absence before the shutdown starts. If we have industry questions, we will default them to Dr. Shapiro where this is possible. We will continue to develop locally, and if a shutdown occurs while we are in the client integration stage, we will simply have to wait until the government reopens to continue integration and demonstrate our software through a local server for purposes of this class. If we are in user testing, we will start with Academic (Dr. Shapiro) and Novice (Lily) user testing, and seek other Industry Professionals for testing through Dr. Shapiro and Lily's contacts if the shutdown is prolonged after those testing rounds.

B.

a. **Issue: Problems with DOE Servers**
b. Fallback plan: We will work on / demonstrate our software through a local server for purposes of this class until the DOE Server is remediated.

C.

a. **Issue: Team Member becomes ill or has an emergency**
b. Fallback plan: If the temporary situation lasts only a couple days, we can delay the deadline of a sprint to allow the team member to finish their portion. If a situation lasts longer, we will reassign tasks within the current sprint.

D.

a. **Issue: Team Member leaves the class or team**
b. Fallback plan: We will reassign tasks within the current sprint and not assign tasks to that team member in future sprints.

E.

a. **Issue: Team Member fails to accomplish their assigned task(s) in a reasonable time frame**
b. Fallback plan: If this causes a bottleneck and is due to a tough question/problem, we will pull the 'Andon cord' and swarm the problem as a team to help out our team member until the issue is resolved. If this is due to a team member's inability to solve the problem, we will reassign the task to another team member. If this is due to lack of effort or otherwise laziness, we will reassign the task to another team member and on the next survey, the team member will receive a lower score.

F.

a. **Issue: Application is running very slowly**
b. Fallback plan: See if the internet connection is stable and that there are no issues on the servers. If everything is working fine, go through the code and refine it to make it more efficient as well as find what could be causing the code to be significantly slow.

G.

a. **Issue: Behind on schedule with the sprints. Velocity is lower than expected.**
b. Fallback plan: If this is caused by a bottleneck, pull the 'Andon cord' and swarm the problem(s) so that the sprint may be completed as soon as possible. We have allowed leeway time (see Leeway / Additional Feature Time) so that Sprint due dates may be extended as needed. Determine why we were not able to complete a sprint on time and plan to fix that in preparation for the next sprint.

H.

a. **Issue: Team Member loses a section of their code**
b. Fallback plan: Have team member revert to their last pushed commit. If the section is large, they have not been pushing their code in a timely manner. Shame them. Then, help them rewrite the portions they lost if their mistake will delay the completion of a sprint. Lecture on good git practices.

I.

a. **Issue: Client contact(s) will not be available for a specific amount of time or is not responding to emails**
b. Fallback plan: If the absence is planned (i.e. leave), we will get contact info for alternate DOE contacts and/or get answers to as many of our questions as we can before the leave starts. In cases where the client is not responding or in the case of unplanned leave, we will continue to work on what we can, and make assumptions (directed by our Academic and Novice users) as needed.

---

11. Probable Technical Requirements

Organization Tools

Slack - official communication

Messenger - informal communication

Trello - kanban board

Google Drive - proposal, documents and documentation

GitHub - share that hot code

Development Tools

Database - SQL

Front End - React.js/ React Redux and CSS or Semantic UI for styling, axios/ajax for calls to backend and database

Backend - PHP, Java with Spring, or Python with Django

IDE's/Environments

Web app targeted for Firefox and Microsoft browser of client choice, pending client approval

Visual Studio Code or Eclipse for IDE

12. System Components Diagram