Overpowered

Summer 2018

Walktime Error

Cynthia Khan, Ricky Le, Caio Melo, Sean Silva, Noah Yasarturk

7/21/18

Table of Contents

Project Proposal

For our project we assume to be working towards the need of an imaginary corporate client. The client receives daily utility bills for several of its buildings and use the service of a third party vendor that parses through the bills and composes the relevant information into a csv file. The csv file is then made available in a sftp server each morning for the client to retrieve. We'll be fabricating this data for purposes of demonstrating the utility of our application using publicly available information as a baseline.

We will develop a web application which will automatically retrieve the csv file and store it to a location on the client's own server daily. The program will then populate/update the client's database with the new information obtained from the files each day. The program will provide a visualization dashboard based on the information garnered from the client's utility bill database for ease of analysis. Using machine learning, the program will process this information further and determine if there is a disproportionate usage of energy by a particular building on a particular day. Should this be true, the program will notify the client via email so that the client can identify the source of the discrepancy and thus control the overconsumption of power. To create a simulation for the software we will be using public datasets from the following website: https://www.data.gov/energy/. These datasets contain both commercial and residential building energy consumption information per hour over a period of years which will be valuable data for our project.

We'll use the Philips Hue Smart Bulb and Bridge system. We may be able to add a feature into the program whereby the dashboard of the program could elucidate if any lights were left on overnight or in a vacant office space so that the user/system can shut off the light remotely. Similarly we can add remote climate control for smart thermostats if we can access an associated API to continue to further contribute to the user's control over utility usage, thus providing the client with a complete cost effective and environment friendly software.

Collaboration

GitHub link: https://github.com/overpowered-gsu
Slack Logs link: https://github.com/overpowered-gsu/slacklogs
YouTube Channel link: https://bit.ly/2JWhv3W
Google Drive link: http://bit.ly/2KyHkDH
SmartSheet link: http://bit.ly/2L3bcfB

2.1 Team Organization

Given the ease with which we have been able to concur on viable project ideas and to acknowledge each other's expertise in varying subfields, we will operate as a democracy.

2.2 Resumes

Cynthia Khan: Graduating Senior in the department of Computer Science at Georgia State University. I have selected Database & Knowledge Systems as my concentration and I find great interest in Data Science and System Automation topics. I have interned at Axiall Chemicals in Summer 2017 and was mentored by a Senior Database Administrator where I got exposure and hand-on experience on MS SQL Server databases. I currently work at Datascape, where I am modelling a visualization network, connecting their data to dynamic dashboards and generating reports from it. Outside of work I have also engaged myself in various voluntary projects relating to the concepts of Big Data, Data Mining and Machine Learning. I am confident in meeting the team's needs for Backend Development.

Ricky Le: I am a Senior at Georgia State University currently pursuing a Bachelor's Degree in Computer Science. I am undeclared in my concentration, however I have a strong interest in Hardware Systems. I am comfortable programming in Java, C, and UNIX Shell, and have a basic understanding of Python, C++, and Assembly (MASM). I have worked at the Mobile Cyber Physical Systems Lab as an intern doing research and development on Backscatter VLC on IoT Systems, and have experience working with microcontrollers, specifically Arduinos. My interest leans towards Electrical/Computer Engineering topics, specifically embedded systems and the Internet of Things.

Caio Melo: I am a Senior at Georgia State University pursuing a B.S. in Computer Science. I have experience with Java and Microsoft Office products. I'm currently in a marketing internship at Beecher Carlson, and I volunteer with Amazon's Mechanical Turks as a side income. I have beginner experience with MASM, Python, and HTML5. AI and botnet technologies fascinate me.

Sean Silva: I am a Senior at Auburn University who is currently pursuing a Bachelor's Degree in Software Engineering. My experience as far as programming goes is an adequate skill set with focus in Java, C++, and Python. I have worked minorly with other languages such as Ruby, Scheme and Prolog. Currently I am fascinated with game design and development as well as the robotics field. I have completed not only class assignments using Unity, but a couple personal projects as well. I currently work with a branch of my university's Office of Information Technology as a multimedia support member. From this work I have experience solving general IT problems that may arise in the audio and visual equipment of the classrooms. When it comes to my specific concentration, I am undecided as to what specific area of software engineering I want to focus on.

Noah Yasarturk: I am a Senior and Presidential Scholar at Georgia State University pursuing a B.S. in Computer Science with a concentration in Database & Knowledge Based Systems with a Math minor. I intend on pursuing a Master of Science in Analytics or Bioinformatics. I am proficient in R and Java primarily but am capable of using Python, C, and MASM (x86). I have a wealth of experience constructing UI in both Java via the Swing package and in R. I have experience with text data mining algorithms as well as creating and analyzing a Twitter stream. I've worked as an undergraduate assistant for Dr. Cox here at GSU's Institute of Neuroscience investigating dendrite morphology in *Drosophila melanogaster*.

Member	Resume Link	Role
Cynthia Khan	https://www.slideshare.net/slideshow/embed_code/key/ban1T3yJekI5kt	Database Developer and Admin
Ricky Le	https://bit.ly/2Jnvmw2	Firmware Developer, API Admin
Caio Melo	https://docs.google.com/document/d/1D26zf54F7E-CJOgTXkIHP2z8GMXVUS4b-jf9V8lSctM/edit?usp=sharing	Lead Web Developer (mainly Front-end)
Sean Silva	https://drive.google.com/file/d/1V4Npqui_jgomTysXCcLnWGy9OwTvR_CO/view?usp=sharing	Lead UI Developer, IT Support, Video Editor
Noah Yasarturk	https://docs.google.com/document/d/1xrt3Hj4Y99Lm7tYyKCDH8wNh2-Ogmmh6qaS-4iFjEV8/edit?usp=sharing	Team Coordinator Junior UI Developer, Assistant Web Developer (mainly back-end)

2.3 Challenges and Risks

Given that we are all college students and furthermore do not live on campus, we all have varying schedules and limited time to dedicate to the project. Due to this fact, coordination and communication will present our greatest challenge.

Risk	Mitigation/Solution
Staff changes- there is a possibility a team member will withdraw from the course or be rendered unable to complete their assigned role; the person we lose may have an irreplaceable skillset	Maintain constant communication so as to ensure we're all aware of each others' responsibilities should they need to be redistributed; ensure for all highly-specialized skills that there is at least 2 people with an understanding of them, even if there is disproportion in knowledge
Dependency- some tasks are going to be dependent upon the completion of other tasks, such as being able to fabricate data before we can process it via machine learning	Use Jira/TSV to Map out general timeline of when Tasks need to be done / which tasks depend on the other
Integration- while the front-end and the back-end will be developed in parallel, if the two development teams are not in frequent communication with another about how their product is being developed, we may be faced with a situation where a back-end feature has nothing on the front-end to access it or a front-end feature is developed without necessary back-end input	Frequent communication between backend development and front end web service. Actively update GitHub with updates on increments of program/code to confirm compatibility.
Learning curve- given that this project may require a few of us to learn to use new tools (such as HTML-heavy web development or the Philips Hue API), it may mean that we spend longer than estimated on any given task	Start practicing languages/technologies we foresee being used and to more accurately estimate the time needed. Invest time ahead in learning
Data- there is a possibility that there is not publicly available data that we can use to train our machine learning model and get an idea of parameters with which we can fabricate our data	We will research available data and data simulation models along with a possibly new project idea

2.4 Monitoring and Reporting Mechanisms

We'll make use of Smartsheet software to track our progress. Additionally, each morning we'll have a Scrum-like report via Slack where we each answer the following questions:

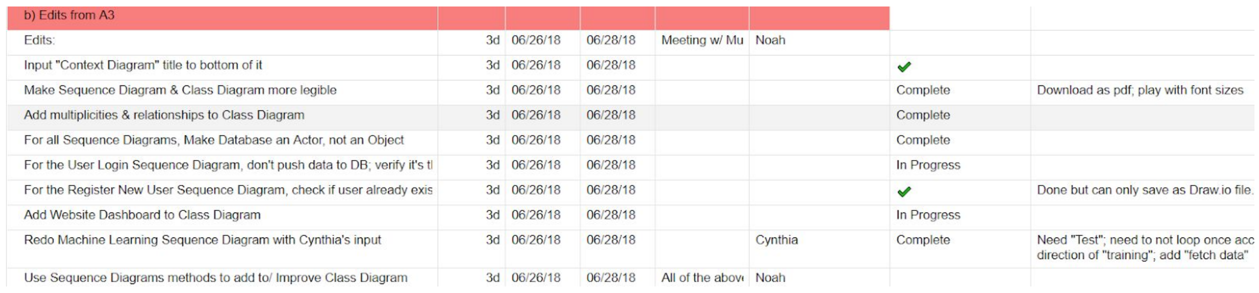
- What did you do yesterday?
- What will you do today?
- What is blocking progress?

Planning and Scheduling

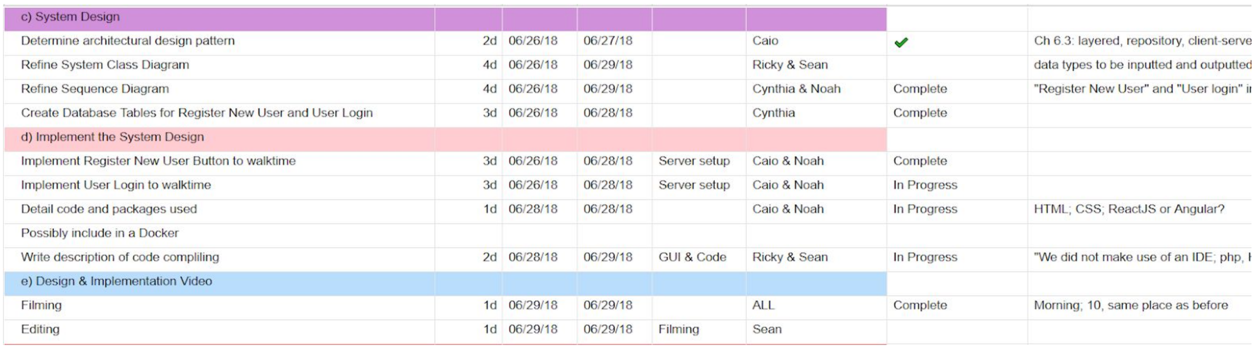
Task	**Effort** (person-days)	**Duration** (hours)	**Dependencies**	**Person**	**Deadline**
Database Setup	1-2	6	Server setup Data	Cynthia Khan	6/25
Learn Tableau	1-2	5	Database and Website setup	Noah Yasarturk, Caio Melo, Sean Silva	6/24
Website Template	2-3	5	Server setup	Caio Melo	6/25
Smart Bulb API Integration	1-2	5-10	Smart Bulb Delivery Time, API Learning Curve	Ricky Le, Noah Yasarturk	6/26
Website Functions Integration	4	4	Functional database, Smartbulb API, Dashboard display	Noah Yasarturk, Sean Silva, Ricky Le, Cynthia Khan	7/12
Dashboard Creation	4	8-12	Database being created; Machine learning	Sean Silva, Noah Yasarturk	7/9

			being able to cypher the data; learning Tableau		
User Registry	1-2	5	Website (front-end/back-end) being functional	Cynthia Khan, Ricky Le, Caio Melo	7/8
Prediction Modelling Integration	2-3	7	Acquire test data and training dataset	Cynthia Khan, Ricky Le, Noah Yasaturk	7/3
Realistic Data Simulation	2-3	8	Power Company Client Research	Noah Yasaturk, Sean Silva	7/1

3.1 Schedule 6/25-6/29 via SmartSheet

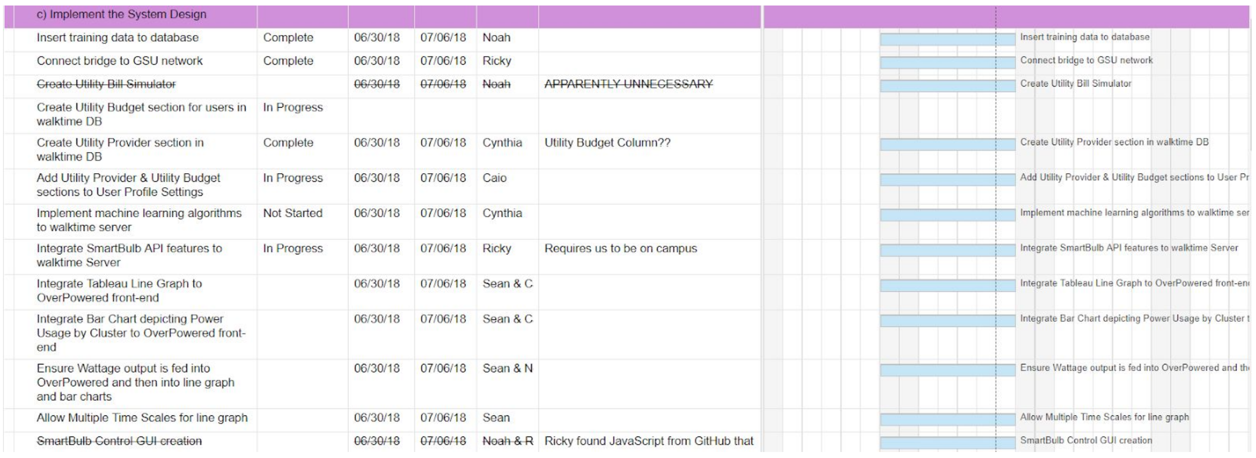


b) Edits from A3							
Edits:	3d	06/26/18	06/28/18	Meeting w/ Mu	Noah		
Input "Context Diagram" title to bottom of it	3d	06/26/18	06/28/18			✓	
Make Sequence Diagram & Class Diagram more legible	3d	06/26/18	06/28/18			Complete	Download as pdf; play with font sizes
Add multiplicities & relationships to Class Diagram	3d	06/26/18	06/28/18			Complete	
For all Sequence Diagrams, Make Database an Actor, not an Object	3d	06/26/18	06/28/18			Complete	
For the User Login Sequence Diagram, don't push data to DB; verify it's tl	3d	06/26/18	06/28/18			In Progress	
For the Register New User Sequence Diagram, check if user already exis	3d	06/26/18	06/28/18			✓	Done but can only save as Draw.io file.
Add Website Dashboard to Class Diagram	3d	06/26/18	06/28/18			In Progress	
Redo Machine Learning Sequence Diagram with Cynthia's input	3d	06/26/18	06/28/18		Cynthia	Complete	Need "Test"; need to not loop once acc direction of "training"; add "fetch data"
Use Sequence Diagrams methods to add to/ Improve Class Diagram	3d	06/26/18	06/28/18	All of the abov	Noah		

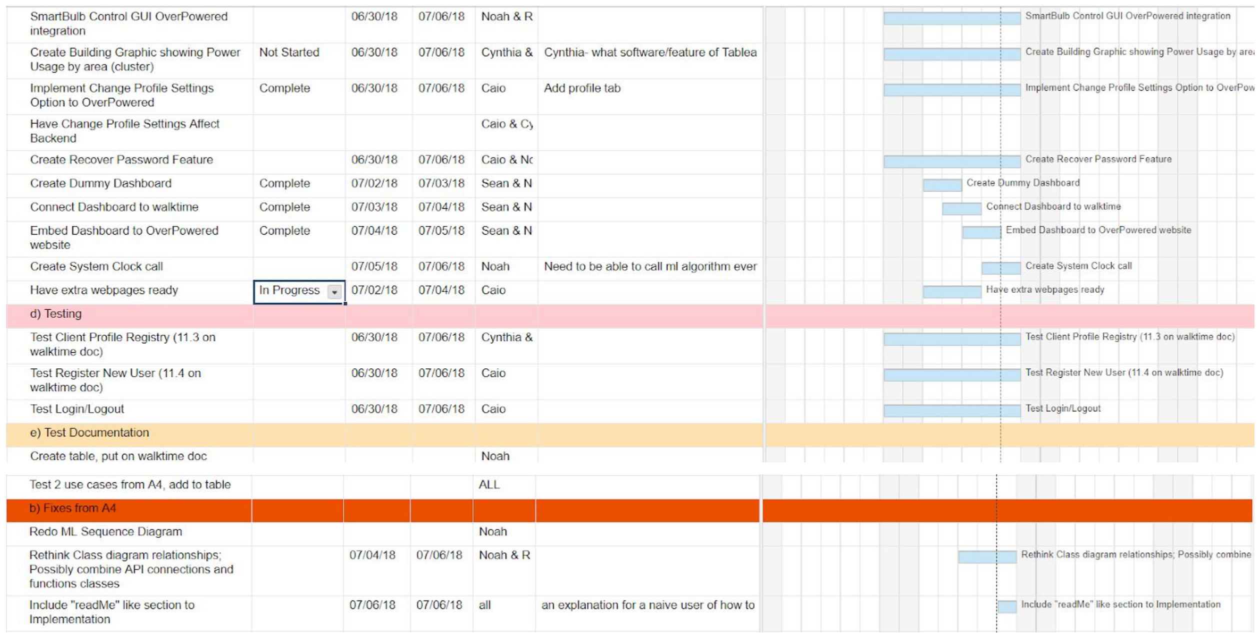


c) System Design							
Determine architectural design pattern	2d	06/26/18	06/27/18		Caio	✓	Ch 6.3: layered, repository, client-serve
Refine System Class Diagram	4d	06/26/18	06/29/18		Ricky & Sean		data types to be inputted and outputted
Refine Sequence Diagram	4d	06/26/18	06/29/18		Cynthia & Noah	Complete	"Register New User" and "User login" i
Create Database Tables for Register New User and User Login	3d	06/26/18	06/28/18		Cynthia	Complete	
d) Implement the System Design							
Implement Register New User Button to walktime	3d	06/26/18	06/28/18	Server setup	Caio & Noah	Complete	
Implement User Login to walktime	3d	06/26/18	06/28/18	Server setup	Caio & Noah	In Progress	
Detail code and packages used	1d	06/28/18	06/28/18		Caio & Noah	In Progress	HTML; CSS; ReactJS or Angular?
Possibly include in a Docker							
Write description of code compliling	2d	06/28/18	06/29/18	GUI & Code	Ricky & Sean	In Progress	"We did not make use of an IDE; php, I
e) Design & Implementation Video							
Filming	1d	06/29/18	06/29/18		ALL	Complete	Morning; 10, same place as before
Editing	1d	06/29/18	06/29/18	Filming	Sean		

3.2 Schedule 7/2-7/6 via SmartSheet

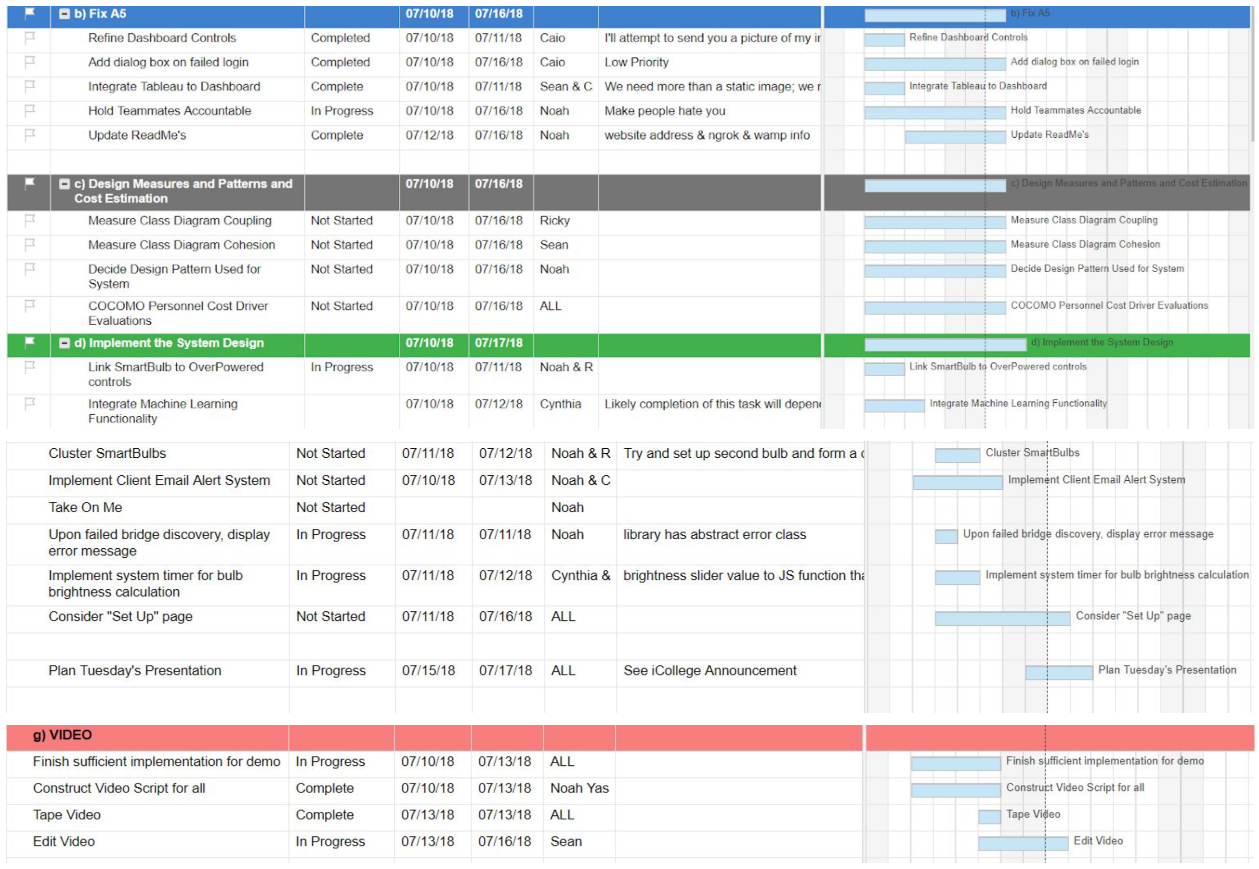


c) Implement the System Design						
Insert training data to database	Complete	06/30/18	07/06/18	Noah		Insert training data to database
Connect bridge to GSU network	Complete	06/30/18	07/06/18	Ricky		Connect bridge to GSU network
~~Create Utility Bill Simulator~~		~~06/30/18~~	~~07/06/18~~	~~Noah~~	~~APPARENTLY UNNECESSARY~~	Create Utility Bill Simulator
Create Utility Budget section for users in walktime DB	In Progress					
Create Utility Provider section in walktime DB	Complete	06/30/18	07/06/18	Cynthia	Utility Budget Column??	Create Utility Provider section in walktime DB
Add Utility Provider & Utility Budget sections to User Profile Settings	In Progress	06/30/18	07/06/18	Caio		Add Utility Provider & Utility Budget sections to User Pr
Implement machine learning algorithms to walktime server	Not Started	06/30/18	07/06/18	Cynthia		Implement machine learning algorithms to walktime ser
Integrate SmartBulb API features to walktime Server	In Progress	06/30/18	07/06/18	Ricky	Requires us to be on campus	Integrate SmartBulb API features to walktime Server
Integrate Tableau Line Graph to OverPowered front-end		06/30/18	07/06/18	Sean & C		Integrate Tableau Line Graph to OverPowered front-en
Integrate Bar Chart depicting Power Usage by Cluster to OverPowered front-end		06/30/18	07/06/18	Sean & C		Integrate Bar Chart depicting Power Usage by Cluster t
Ensure Wattage output is fed into OverPowered and then into line graph and bar charts		06/30/18	07/06/18	Sean & N		Ensure Wattage output is fed into OverPowered and th
Allow Multiple Time Scales for line graph		06/30/18	07/06/18	Sean		Allow Multiple Time Scales for line graph
~~SmartBulb Control GUI creation~~		~~06/30/18~~	~~07/06/18~~	~~Noah & R~~	Ricky found JavaScript from GitHub that	SmartBulb Control GUI creation



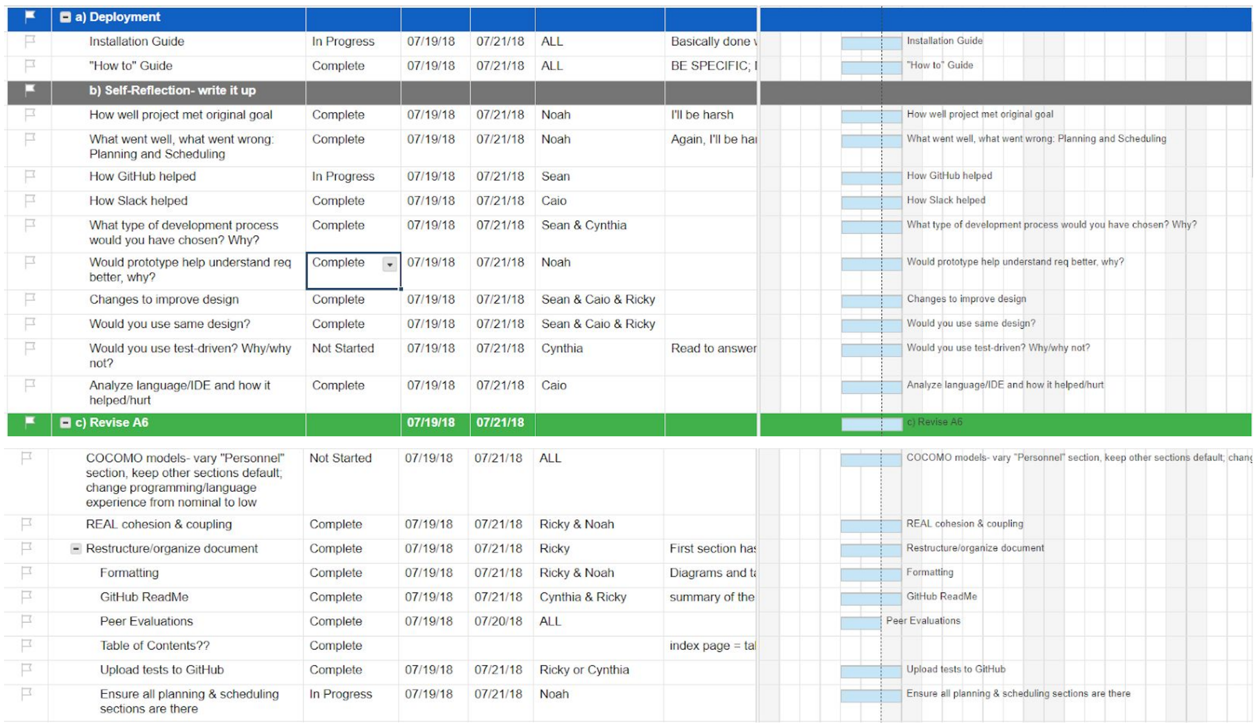
SmartBulb Control GUI OverPowered integration		06/30/18	07/06/18	Noah & R	
Create Building Graphic showing Power Usage by area (cluster)	Not Started	06/30/18	07/06/18	Cynthia &	Cynthia- what software/feature of Tablea
Implement Change Profile Settings Option to OverPowered	Complete	06/30/18	07/06/18	Caio	Add profile tab
Have Change Profile Settings Affect Backend				Caio & Cy	
Create Recover Password Feature		06/30/18	07/06/18	Caio & Nc	
Create Dummy Dashboard	Complete	07/02/18	07/03/18	Sean & N	
Connect Dashboard to walktime	Complete	07/03/18	07/04/18	Sean & N	
Embed Dashboard to OverPowered website	Complete	07/04/18	07/05/18	Sean & N	
Create System Clock call		07/05/18	07/06/18	Noah	Need to be able to call ml algorithm ever
Have extra webpages ready	In Progress	07/02/18	07/04/18	Caio	
d) Testing					
Test Client Profile Registry (11.3 on walktime doc)		06/30/18	07/06/18	Cynthia &	
Test Register New User (11.4 on walktime doc)		06/30/18	07/06/18	Caio	
Test Login/Logout		06/30/18	07/06/18	Caio	
e) Test Documentation					
Create table, put on walktime doc				Noah	
Test 2 use cases from A4, add to table				ALL	
b) Fixes from A4					
Redo ML Sequence Diagram				Noah	
Rethink Class diagram relationships; Possibly combine API connections and functions classes		07/04/18	07/06/18	Noah & R	
Include "readMe" like section to Implementation		07/06/18	07/06/18	all	an explanation for a naive user of how to

3.3 Schedule 7/8-7/16 via SmartSheet



b) Fix A5		**07/10/18**	**07/16/18**		
Refine Dashboard Controls	Completed	07/10/18	07/11/18	Caio	I'll attempt to send you a picture of my ir
Add dialog box on failed login	Completed	07/10/18	07/16/18	Caio	Low Priority
Integrate Tableau to Dashboard	Complete	07/10/18	07/11/18	Sean & C	We need more than a static image; we r
Hold Teammates Accountable	In Progress	07/10/18	07/16/18	Noah	Make people hate you
Update ReadMe's	Complete	07/12/18	07/16/18	Noah	website address & ngrok & wamp info
c) Design Measures and Patterns and Cost Estimation		**07/10/18**	**07/16/18**		
Measure Class Diagram Coupling	Not Started	07/10/18	07/16/18	Ricky	
Measure Class Diagram Cohesion	Not Started	07/10/18	07/16/18	Sean	
Decide Design Pattern Used for System	Not Started	07/10/18	07/16/18	Noah	
COCOMO Personnel Cost Driver Evaluations	Not Started	07/10/18	07/16/18	ALL	
d) Implement the System Design		**07/10/18**	**07/17/18**		
Link SmartBulb to OverPowered controls	In Progress	07/10/18	07/11/18	Noah & R	
Integrate Machine Learning Functionality		07/10/18	07/12/18	Cynthia	Likely completion of this task will depen
Cluster SmartBulbs	Not Started	07/11/18	07/12/18	Noah & R	Try and set up second bulb and form a c
Implement Client Email Alert System	Not Started	07/10/18	07/13/18	Noah & C	
Take On Me	Not Started			Noah	
Upon failed bridge discovery, display error message	In Progress	07/11/18	07/11/18	Noah	library has abstract error class
Implement system timer for bulb brightness calculation	In Progress	07/11/18	07/12/18	Cynthia &	brightness slider value to JS function tha
Consider "Set Up" page	Not Started	07/11/18	07/16/18	ALL	
Plan Tuesday's Presentation	In Progress	07/15/18	07/17/18	ALL	See iCollege Announcement
g) VIDEO					
Finish sufficient implementation for demo	In Progress	07/10/18	07/13/18	ALL	
Construct Video Script for all	Complete	07/10/18	07/13/18	Noah Yas	
Tape Video	Complete	07/13/18	07/13/18	ALL	
Edit Video	In Progress	07/13/18	07/16/18	Sean	

3.4 7/18 - 7/21 via SmartSheet



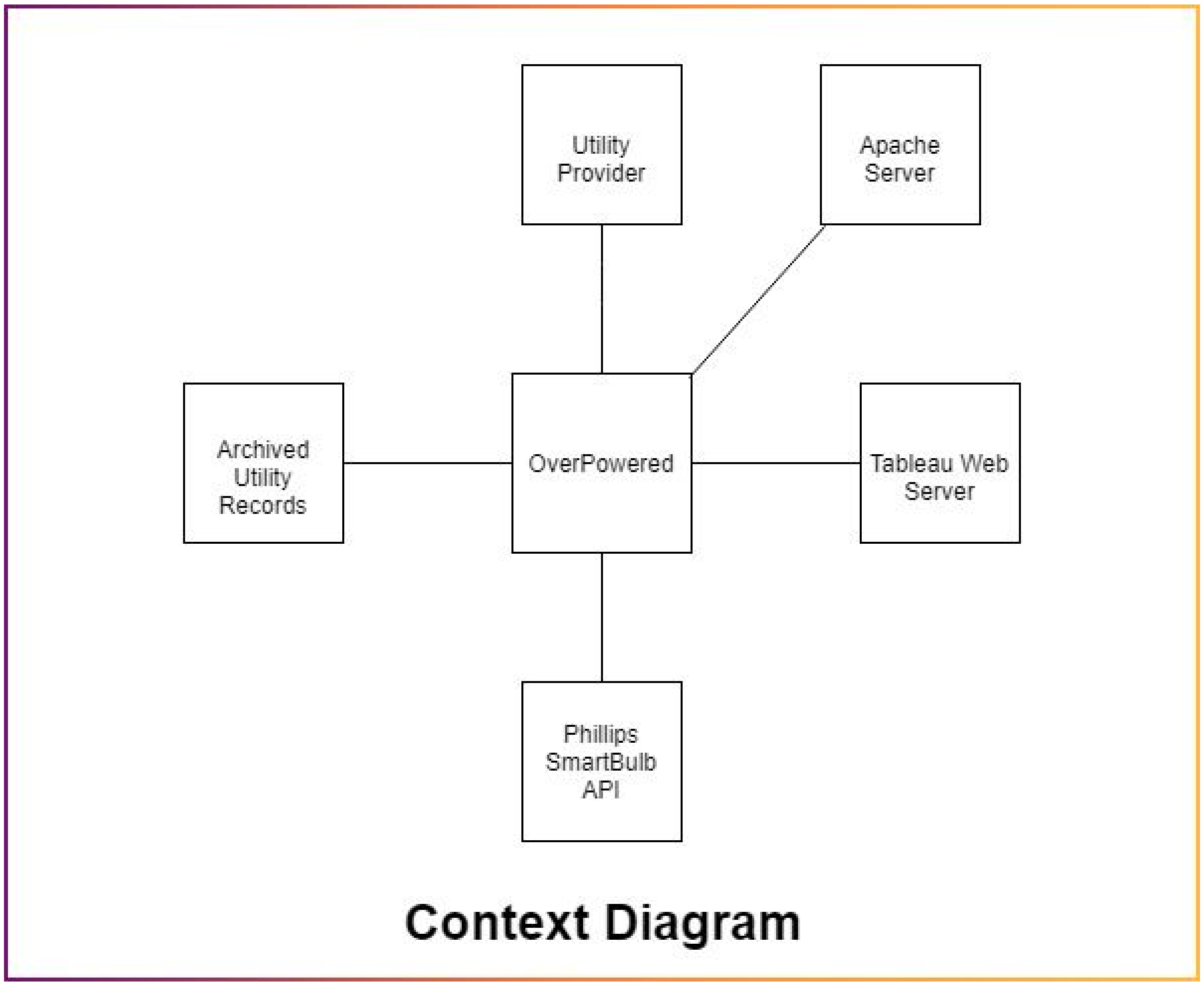
a) Deployment						
Installation Guide	In Progress	07/19/18	07/21/18	ALL	Basically done	Installation Guide
"How to" Guide	Complete	07/19/18	07/21/18	ALL	BE SPECIFIC;	"How to" Guide
b) Self-Reflection- write it up						
How well project met original goal	Complete	07/19/18	07/21/18	Noah	I'll be harsh	How well project met original goal
What went well, what went wrong: Planning and Scheduling	Complete	07/19/18	07/21/18	Noah	Again, I'll be ha	What went well, what went wrong: Planning and Scheduling
How GitHub helped	In Progress	07/19/18	07/21/18	Sean		How GitHub helped
How Slack helped	Complete	07/19/18	07/21/18	Caio		How Slack helped
What type of development process would you have chosen? Why?	Complete	07/19/18	07/21/18	Sean & Cynthia		What type of development process would you have chosen? Why?
Would prototype help understand req better, why?	Complete	07/19/18	07/21/18	Noah		Would prototype help understand req better, why?
Changes to improve design	Complete	07/19/18	07/21/18	Sean & Caio & Ricky		Changes to improve design
Would you use same design?	Complete	07/19/18	07/21/18	Sean & Caio & Ricky		Would you use same design?
Would you use test-driven? Why/why not?	Not Started	07/19/18	07/21/18	Cynthia	Read to answer	Would you use test-driven? Why/why not?
Analyze language/IDE and how it helped/hurt	Complete	07/19/18	07/21/18	Caio		Analyze language/IDE and how it helped/hurt
c) Revise A6		**07/19/18**	**07/21/18**			c) Revise A6
COCOMO models- vary "Personnel" section, keep other sections default; change programming/language experience from nominal to low	Not Started	07/19/18	07/21/18	ALL		COCOMO models- vary "Personnel" section, keep other sections default; chang
REAL cohesion & coupling	Complete	07/19/18	07/21/18	Ricky & Noah		REAL cohesion & coupling
Restructure/organize document	Complete	07/19/18	07/21/18	Ricky	First section ha	Restructure/organize document
Formatting	Complete	07/19/18	07/21/18	Ricky & Noah	Diagrams and t	Formatting
GitHub ReadMe	Complete	07/19/18	07/21/18	Cynthia & Ricky	summary of the	GitHub ReadMe
Peer Evaluations	Complete	07/19/18	07/20/18	ALL		Peer Evaluations
Table of Contents??	Complete				index page = ta	
Upload tests to GitHub	Complete	07/19/18	07/21/18	Ricky or Cynthia		Upload tests to GitHub
Ensure all planning & scheduling sections are there	In Progress	07/19/18	07/21/18	Noah		Ensure all planning & scheduling sections are there

Problem Statement

Overpowered is an intelligent system by which business clients can reduce their utility bill through provided data analytics and integrated hardware systems. It is for business clients and/or universities that can make widespread use of IoT technology (smart bulbs in particular) for more efficient power-gauging and monitoring. Overpowered solves the problem of power over-consumption, making it environmentally friendly, and excessive spending on utility bills and it drastically reduces the time clients spend analyzing their utility bills. While there are service alternatives such as motion sensor lights, our service provides a more reliable way to monitor light usage. Overpowered is a compelling project to pursue due to the fact that it incorporates new, emerging technology (IoT Technology) with machine learning concepts and data analytics to solve an issue that is applicable to a large group of people/businesses. This project's top level objective would be to have complete adoption of the system by a major consumer of electricity such as a residential building, or a college campus. This project incorporates machine learning with already available data to determine even the slightest changes that will benefit the energy consumption of a building as a whole. It will also provide a more secure system, as our database would be hosted at the client site, and not through the cloud. Target customers for this project include businesses with large buildings who receive daily updates of their utility usage and make widespread use of smart bulb technology. The scope of our project encompasses any building with multiple stories and/or multiple buildings of the same caliber. There are plenty of research papers published that reference using machine learning to predict and model energy consumption within buildings and how to make them efficient, however they have no systems in place to actually implement them. Upon further research, a company named *Verdigris* has implemented a similar product as our project, however this company focuses more on the hardware elements of energy management. They store their user information within a cloud database, which would not be as secure as an adhoc physical server.

System Requirements

Context Model



5.1 Requirement #: 1

Use Case: N/A
Date: June 15th, 2018
Introduction: Obtain Utility Bill Data and Implement into Database
Rationale: Information is needed to feed our machine learning system in order for it to function properly.

Author: Walktime Error
Input: Public Utility Usage Data
Requirement Description: Using public data of utility usage of specific buildings and businesses, the information fed to the system will allow precise and accurate predictions for optimal to suboptimal utility usage and spending.
Outputs: Functional Machine Learning System that is Precise and Accurate
Related Requirements: 2, 8

5.2 Requirement #: 2

Use Case: Database Functionality
Date: June 15th, 2018
Introduction: Create/Configure Server to be Database
Rationale: A Database needs to be created in order to implement machine learning algorithms, as well as a User Registry.
Author: Walktime Error
Input: User inputted Utility Data, Profile information
Requirement Description: Server access needs to be obtained to create a Database. This database will serve as the hub for our machine learning process, as well as our user registry. It will contain user profile information, as well as user inputted utility data.
Outputs: Machine Learning Data Visualization, User Registry, Email Alert System
Related Requirements: 1, 3, 4, 6, 8

5.3 Requirement #: 3

Use Case: Register New User, User Login, Database Functionality, Smartbulb API Functionality
Date: June 21st, 2018
Introduction: Develop Client Registry Parameters and Definition
Rationale: User needs to be able to register a profile in order to limit access to Utility Administrators, and to submit information to be analyzed and produce visualized data, such as charts, graphs, and graphics.
Author: Walktime Error
Input: Username, Password, Building Layout Schematics, Utility Admin Email, Utility Provider, Utility Bill Budget Target
Requirement Description: User submits Username, Password and Full Name, which is then placed in corresponding table within database. User presses next button which then leads to submitting utility administrator email, utility provider, utility bill budget target, building ID, and number of floors. User then presses the next button which then allows user to register smart devices. After registration is complete, user is then able to define clusters/groups from registered devices, and link them to the corresponding building layout data.

Outputs: Profile information is saved into Database, which allows future login to access website dashboard.
Related Requirements: 2, 4, 5, 7

5.4 Requirement #: 4

Use Case: Register New User/Login
Date: June 20th, 2018
Introduction: User Credential setup and execution.
Rationale: Create a website that allows users to submit information in order to receive data analytics and control over smart devices to decrease utility spending.
Author: Walktime Error
Input:
- Username, Password

Requirement Description: Website should have the following functions:
- User Profile Registration
- User Login
- Remote Smart Device Control
- Access to Data Analytics
- User Logout
- Backend integration of Database

Outputs: A website with working buttons that lead to their corresponding functions.
Related Requirements: 2, 3, 5, 7, 8

5.5 Requirement #: 5

Use Case: Recover Password
Date: June 20th, 2018
Introduction: Recover lost password
Rationale: Allow users to recover lost password
Author: Walktime Error
Input:
- Email Address

Requirement Description: Recovery Portal should:
- Prompt user for email address used to register
- Send Credentials to email provided.

Outputs: Send recovery information to provided email.
Related Requirements: 4

5.6 Requirement #: 6

Use Case: Dashboard Display Functionality
Date: June 15th, 2018
Introduction: Website Aesthetics / Dashboard
Rationale: Along with integration of website functionality, the website must look simple and easy to use, and not confusing to user.
Author: Walktime Error
Input: Website Functions
Requirement Description: All backend functions should be tied to frontend functions, including the use of sliders, buttons, and aesthetic animations for smoothness of use.
Outputs: An aesthetically pleasing, easy to use website.
Related Requirements: 3, 4, 7

5.7 Requirement #: 7

Use Case: Database Display Functionality, Smart Bulb API Functionality, Machine Learning Functionality
Date: June 15th, 2018
Introduction: Implement Machine Learning Algorithms
Rationale: Using this allows our system to be intelligent and provide accurate utility predictions to better optimize utility efficiency
Author: Walktime Error
Input: Utility Data from Database
Requirement Description: Using public Machine Learning packages and algorithms, the system will use client provided utility data to analyze and identify inefficient utility use per building and send out an email whenever inefficiency is identified. Inefficiency is defined through the mathematics behind the machine learning algorithms.
Outputs: Alert Email , Machine Learning Predictions, Data visualizations
Related Requirements: 2, 7, 8

5.8 Requirement #: 8

Use Case: Dashboard Display Functionality, Smart Bulb API Functionality
Date: June 21st, 2018
Introduction: Configure/Integrate Smart Bulb API Features
Rationale: Using the Philips Hue Smart Bulb API, integrate the smart devices functions with website functions to have more utility control, as well as easier access, as all needed functions will be provided in a single area.
Author: Walktime Error
Input: Smart Bulb API Functions

Requirement Description: The Philips Hue API Interface will be integrated with the system website through Node.js application development. Corresponding functions/ desired functions such as remote toggle, remote intensity slider, and wattage output information will be called and implemented within Node.js application, and then linked to output on website. The wattage output information will be placed within database for machine learning purposes.
Outputs: Remote Control of Lighting, Grouping/Clustering of Individual Lights, Timed toggles, Integration within Website Features
Related Requirements: 3, 4, 5, 6

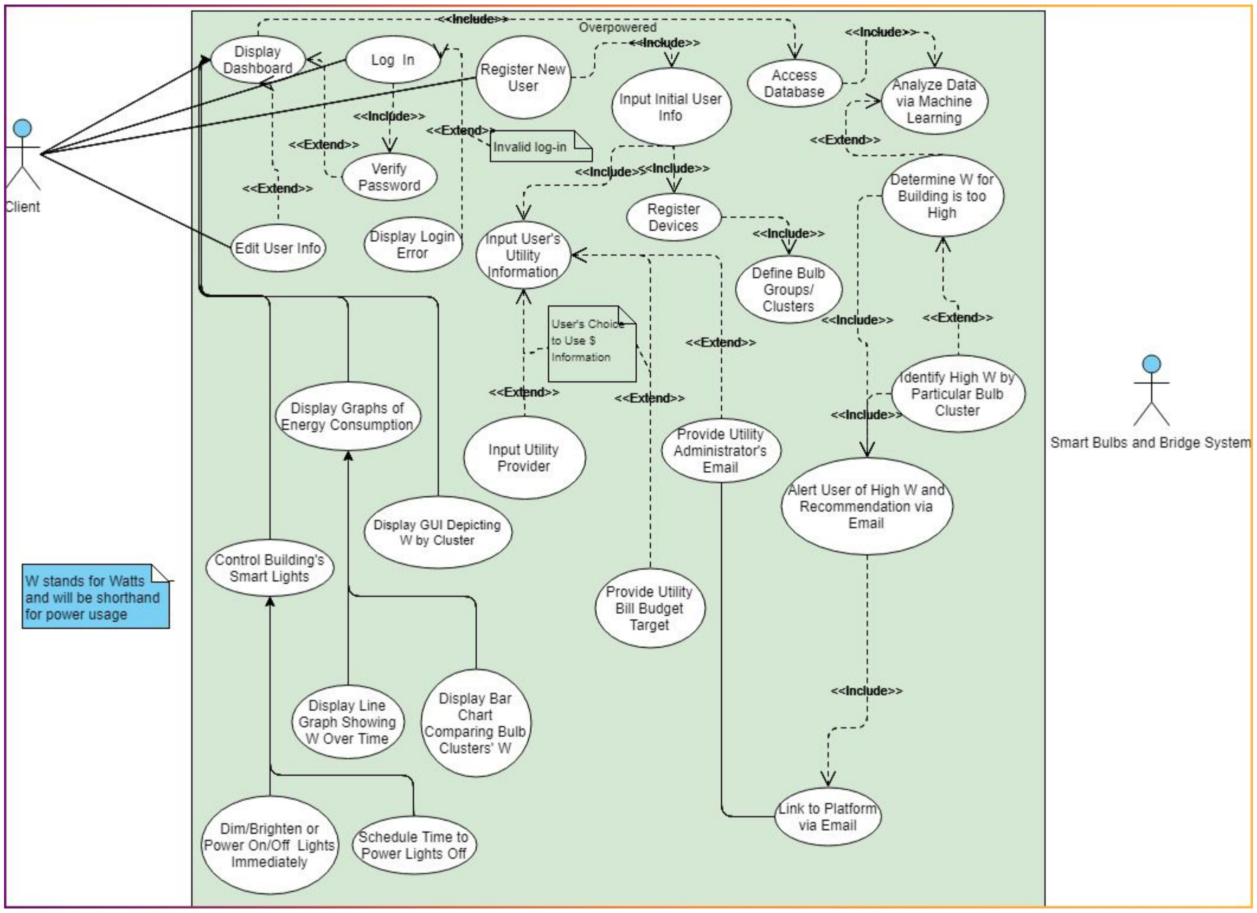
5.9 Requirement #: 9

Use Case: Dashboard Display Functionality, Smart Bulb API Functionality, Machine Learning Functionality
Date: June 15th, 2018
Introduction: Integrate Data Visualization
Rationale: Integration of Data Visualization will allow easier to understand information and data analytics, as it will take information from the machine learning system and incorporate visual data such as graphs, charts, and graphics.
Author: Walktime Error
Input: Database Information Pertaining to Utilities
Requirement Description: Using user submitted information about utility usage, and getting outputs from the machine learning system, the integration will provide visual data to help improve utility spending and efficiency.
Outputs: Charts, Graphs, and Graphics
Related Requirements: 1, 2, 4, 5, 6

Use Cases

6.1 Use Case Diagram

The Use Case Diagram file can be referred to in Appendix 1.2.


Client
Display Dashboard
Log In
Register New User
Overpowered
Input Initial User Info
Access Database
Analyze Data via Machine Learning
Verify Password
Invalid log-in
Register Devices
Determine W for Building is too High
Edit User Info
Display Login Error
Input User's Utility Information
Define Bulb Groups/ Clusters
User's Choice to Use $ Information
Identify High W by Particular Bulb Cluster
Display Graphs of Energy Consumption
Input Utility Provider
Provide Utility Administrator's Email
Alert User of High W and Recommendation via Email
Display GUI Depicting W by Cluster
Control Building's Smart Lights
Provide Utility Bill Budget Target
W stands for Watts and will be shorthand for power usage
Display Line Graph Showing W Over Time
Display Bar Chart Comparing Bulb Clusters' W
Link to Platform via Email
Dim/Brighten or Power On/Off Lights Immediately
Schedule Time to Power Lights Off
Smart Bulbs and Bridge System
<<Include>>
<<Extend>>

6.2 Use Case Descriptions

6.2.1 Use Case Name: Register New User

Summary: Client will be able to register a new user profile and add pertinent information.
Basic Course of Events:

1. Client clicks button to register new user.
2. The page redirects to another page to input basic user information, such as username, password, and Full Name.
3. The information submitted will be saved onto the database for login information.
4. The client clicks next button to submit more information required.
5. The page redirects to another page to input Building Layout, as well as pertinent information related to utilities (Utility Administrator email, utility provider, and

provide a utility bill budget target). The Building Layout will consist of monocolor floor plans / blueprints for machine learning color coding by cluster.

6. Information will be saved onto the database for machine learning data analysis, and used for graphics and energy-use suggestions.
7. The client clicks the next button to submit more information required.
8. The page redirects to another page to register smart devices, and to define clusters/groups. These clusters will later be used to identify specific areas for power usage comparison.
9. Information will be saved and be used for machine learning data analysis, and will provide future functionality on the Dashboard.
10. The client clicks the Done button, and is redirected to the homepage / dashboard.

Alternative Paths: None
Exception Paths: In step 2, if illegal characters are used, or username/password requirements are not fulfilled, the page will reload and will ask to resubmit information without illegal characters/ to fulfill requirements. The submitted password will have a length requirement of at least 8 characters. In step 5, if an unreadable Building Layout format is submitted, or any invalid information pertaining utilities is submitted, the page will reload and ask to re-submit information.
Extension Points: User will be able to go back and edit information submitted through an option on the dashboard
Trigger: The user wants to register a new profile.
Assumptions: The user has access to pertinent utility, smart device, and building layout information.
Precondition: The database has been established and can receive multiple forms of information.
Postcondition: The user has created a new login profile.
Author: Walktime Error
Date: June 15th, 2018

6.2.2 Use Case Name: User Login

Summary: User will be able to login using existing profile information.
Basic Course of Events:

1. User clicks a button to redirect to a login Page.
2. The Page will ask for username and password.
3. User submits username and password.
4. The database will verify the username and password.
5. The page will redirect to website dashboard.

Alternative Paths: If the user does not accurately submit username and/or password after 5 tries, the user is locked from attempting to log in for 30 minutes.
Exception Paths: In step 3, if user does not submit correct username and/or password, the page will reload and ask user to resubmit. (Display login error)
Extension Points: None

Trigger: User wants to log in to use application.
Assumptions: User has already registered a profile and has submitted pertinent information.
Precondition: The database is able to verify login credentials.
Postcondition: The login information has been accepted and the user can now proceed.
Author: Walktime Error
Date: June 15th, 2018

6.2.3 Use Case Name: Dashboard Display Functionality

Summary: After logging in, the user will see and be able to use all dashboard functionalities.
Basic Course of Events:

1. The user successfully logs in, and can see the functions placed on the dashboard.
2. The user has the option to:
    a. Control Building's Smart Lights
    b. Display Graphs of Energy Consumption
    c. Display Building Graphic depicting power usage.
    d. Change profile settings
    e. Log out

Alternative Paths: From Step 2a, the user has the option to dim/brighten or power on/off Lights immediately, whether in defined cluster, or individual bulbs. Defined Clusters and registered smart devices will come from the user's submitted profile information. From Step 2b, user can view graphs or charts of energy consumption, created by the system. User can choose to categorize the information provided by Daily, Weekly, or Monthly, Annually. Information will come from the user's submitted profile information and information stored in the database through a period of time. From Step 2c, the user can view a graphic depicting the Building schematics, displaying power usage per building (color-coded). The building schematics will be retrieved from the submitted building information within the user's profile. From Step 2d, the user can edit submitted information in their profile, change password, or change profile settings. From here, the user can also redefine existent bulb clusters, as well as register new smart devices. The user can then click a button to save edited information, and will return to the Dashboard. From Step 2e, User will be logged out and will no longer be able to access the dashboard interface.
Exception Paths: None
Extension Points: From Steps 2a-d, the user will have access to a button that allows them to return to the dashboard. In regards to Step 2d, pressing this back button will not save edited information and a notification will be generated for the user.
Trigger: The user wants to access dashboard functionalities.
Assumptions: The login has been successful, thus allowing user to access dashboard.
Precondition: The front-end and back-end functionalties have been linked at their appropriate places. Information is able to be edited through the profile settings and the edits will change within the database. The machine learning system has already been set up (gone through all submitted information), and graphics have been produced based off of said information.
Postcondition: User is able to use all dashboard functionalities and is able to log off.

Author: Walktime Error
Date: June 15th, 2018

6.2.4 Use Case Name: Database Functionality

Summary: The Database will store all information pertaining to the energy consumption of client's assets and employee login credentials. Thus it will provide support to the front end applications.

Basic Course of Events:

1. Hourly energy consumption measures from installed meters would be recorded and updated in the database.
2. When a new user registers, the employee's ID, username and password will be saved in the database
3. When a user tries to login, the rows in the database will be searched sequentially to validate the user's credentials
4. When creating a prediction model, the data held in the database will be used in a systematic manner.

Alternative Paths: For Task 1, data collected from installed meters would be inserted to the database through automated scripts. In case, of corrupted data or data not meeting the database integrity, that particular record will be discarded and this action will be recorded in a log along with the reason for discarding the data. For Task 2, user information must meet the requirements specified in the database. Ineligible usernames and/or passwords would be rejected and this will trigger an alert to be sent in the form of a message to the user. For Task 3, information collected from user input would be scanned against all records in the Employee table for a match. In case of a match found a success message would be triggered. In case of no match, a failure message would be triggered for the user. For Task 4, the database would act as a source for the information needed by the Machine learning algorithms.

Exception Paths: Invalid or corrupted data.

Extension Points: Regular scheduled backups setup for the client

Trigger: Any information that needs to be saved or retrieved.

Assumptions: The database schema functions as desired and data is not corrupted.

Precondition: Database schema is setup properly with intended dependencies.

Postcondition: Data can be stored and retrieved with no unexpected delays

Author: Walktime Error

Date: June 15th, 2018

6.2.5 Use Case Name: Smart Bulb API Functionality

Summary: Using the Philips Hue Smart Bulb API, functions will be integrated within the machine learning system, as well as the website.

Basic Course of Events:

1. Setup between the bridge and bulbs, and the network to ensure the devices are functioning properly.
2. Establish a connection between Smart Devices, and the Hue App API.
3. Apply methods within the API to create functions desired for machine learning and website functionality.

Alternative Paths: From step 3, multiple functions need to be created. The first function required is the ability to register devices individually and define clusters. The second function is to give our system access to be able to remotely toggle and change intensity of bulb clusters. The third function is to create a readable and storable format for wattage output of bulb clusters, to implement this information within our machine learning system.

Exception Paths: Bulb information regarding wattage is unreadable to the machine learning system, which function must be re-evaluated to output desirable information. The room/cluster of bulbs have been remotely toggled while there is occupancy, which occupants can manually reset by toggling the wall switch, or through the smartphone app.
Extension Points: User must be able to redefine clusters, as well as register new smart devices.
Trigger: The User would like to remotely toggle lights, and have wattage information available to be compared between clusters and buildings.
Assumptions: The Philips Hue API is open and readily available to use.
Precondition: Access to the Philips Hue API, Places where Smart Bulbs are implemented have a wall switch that remains in the On position.
Postcondition: Desired functions readily working for machine learning process, website functionality, and smart device registration.
Author: Walktime Error
Date: June 15th, 2018

6.2.6 Use Case Name: Machine Learning Functionality

Summary: Provide Machine Learning features to the user such as notifying of unusually high energy consumption in any particular area.
Basic Course of Events:
A substantial amount of training data is stored in the database. Then using a chosen Machine Learning Algorithm, a prediction model is produced. An accuracy score is also derived from the training data. The process is repeated for multiple algorithms until the best suited algorithm with the highest accuracy is obtained. Once the algorithm is selected, the prediction model produced by the algorithm is used to predict future instances. For example, when the system is trained in the described manner into knowing the energy consumption of a building at any particular hour, the next time the system is given the consumption within 15 mins, it would be able to predict if there will be an over consumption, average consumption or low consumption by the end of the hour. This way the system can alert the user so that the user may choose to take action immediately and save precious energy.

Alternative Paths: Every hour, at a certain predetermined time, the system will automatically run a few selected algorithms against the available dataset and determine the best algorithm based on accuracy. Once it has determined the algorithm, it will then store the prediction model and use that model to predict the outcome at the end of the hour. If the prediction is lower than or within the threshold then no action is taken until the next hour when the process is repeated. If the prediction is higher than the threshold then the system triggers an alert and starts the process of notifying the user of the anticipated high consumption and the possible steps that may be taken. After that the system again waits for the next hour to repeat the process.
Exception Paths: None of the algorithms might produce an accurate enough result which may be depended upon.
Extension Points: The user may also be advised on which devices are having the highest consumption rate.
Trigger: A schedule may be set for the system to start this entire process.
Assumptions: The Machine Learning kit functions as desired and produces accurate predictions.
Precondition: The training data is clean and vast enough for optimal results.
Postcondition: The system is able to predict the energy consumption outcome by the end of the hour.
Author: Walktime Error
Date: June 15th, 2018

Test Cases

7.1 Integrate Utility Bills and Database

Requirement Description:	Using public data of utility usage of specific buildings and businesses, the information fed to the system will allow precise and accurate predictions for optimal to suboptimal utility usage and spending.
Rationale:	Information is needed to feed our machine learning system in order for it to function properly.
Inputs:	Public Utility Usage Data.
Outputs:	Usable data stored in the database that will allow a functional Machine Learning System.
Persistent Changes:	Database will be populated with Utility Data, Machine Learning System will be able to function.

Related Requirements and Use Cases:	Database Setup
Test Cases:	1. Open SSMS 2. Connect to Database 3. Open New Query for table 'Machine Learning' 4. Use SELECT Top 100 * from Column_Name 5. Verify that data matches with sample data 6. Verify that column names match ERD

7.2 Database Setup

Requirement Description:	Setup Database server and management tool. Start Database server. Create a new database and connect to it through the management tool. Create tables with designed constraints and columns. Create user accounts for the rest of the development team and grant them admin privilege to the database.
Rationale:	Guarantee we have a functional Database and backend for our system.
Inputs:	None.
Outputs:	Successful connection to the database is established. Tables and column structures are shown as desired.
Persistent Changes:	No change.
Related Requirements and Use Cases:	None
Test Cases::	1. Open SQL Server Management Studio. 2. Connect to the database instance using Windows authentication. 3. Start a new query window. 4. Type in SELECT statement to view the tables and columns in the database. 5. Confirm the all tables contain the columns as per design.

7.2.2 Insert Data into Database

Requirement Description:	Insert Utility data from flat file into the appropriate tables. Retrieve stored data.
Rationale:	Make sure we have the ability to store relevant Utility data in server, and retrieve.
Inputs:	Utility data file
Outputs:	The related tables should be populated with the correct data. Relations between tables should yield correct result through JOIN statements.
Persistent Changes:	New information will be stored in the database, and retrieved.
Related Requirements and Use Cases:	Database and tables are created as per design.
Initialization:	New Query window connected to the appropriate table
Test Cases:	1. Perform SELECT Statements for each table to verify the data in each column 2. Verify the data types for each column 3. Verify primary and secondary key constraints. Verify null constraint with INSERT statement. 4. Verify table relationships with JOIN statement. 5. Verify that passwords are encrypted.

7.2.3 Get Non-empty Form List (Form Storage System)

Requirement Description:	Checks that the saved form has a single form entry.
Rationale:	Guarantees a 1-to-1 relationship with the form and entry.
Inputs:	None
Outputs:	A list of forms is returned. There is exactly one entry in the form list. No exception is thrown.

Related Requirements and Use Cases:	None
Initialization:	The user name is already set in the system.
Test Cases:	1. Save an empty form with Requirement Description “Greg’s data”. No other data in the form is filled in. 2. Retrieve (loads) the form. 3. Verify that the retrieved form only contains one entry. 4. Verify that the entry is an element with an id number and a Requirement Description “Greg’s data”.

7.3 Client Profile Registry

Requirement Description:	Following username and password definition, when the user presses the next button, the user is redirected to a page to input pertinent information for utility usage and building schematics.
Rationale:	Upon signing up for the service, data from the client will be needed for our system to function.
Inputs:	Utility Administrator email, Utility Provider, Utility Bill Budget Target, Building Schematics (building id number, number of floors)
Outputs:	Inputted information goes to the correct respective tables within the database and stored under user profile ID.
Persistent Changes:	The system will then have Data to work with, and populate the machine learning system.
Related Requirements and Use Cases:	Database Setup, Front-end Website Display
Test Cases:	1. Enter utility administrator email, utility provider, utility bill budget target, and building schematics in respective areas and click Next button. 2. Verify that the information submitted has gone to the correct respective tables within database, stored under client’s unique username. 3. Verify that the Next button redirects user to the next information submission page.

7.3.2 Smart Device Registry

Requirement Description:	Following client profile registry information, when the user presses the next button, the user is redirected to a page to register smart bulbs and smart bridge.
Rationale:	Client Registry will allow the Smart Bulb to take in new data.
Inputs:	Smart Bridge IDs, Smart Bulb IDs
Outputs:	All initial smart devices are registered and linked to a unique ID.
Persistent Changes:	This will allow the Smart Bulb access in gaining new information for the Machine Learning aspect.
Related Requirements and Use Cases:	Physical Power and Internet Connection to Smart Bridge, Smart Bulb API Integration
Test Cases:	1. Press link button on Smart Bridge. 2. Verify that Smart Bridge is connected to the internet and is actively searching for devices. 3. After a designated time of searching, the Smart Bridge will detect and list out all Smart Bulbs within vicinity of its range. 4. Verify that all desired smart bulbs have been detected and has unique ID's.

7.3.3 Smart Bulb Cluster Definition

Requirement Description:	After Smart Device Registry, the User will be able to define clusters/groups of bulbs and link them to the corresponding areas given from building schematics.
Rationale:	Allows for better data visualization from Tableau.
Inputs:	Group Names, Building IDs, floor number, designated areas on floor
Outputs:	Smart Bulbs and Bridges are registered and defined under a user-defined group name, and linked to building ID number and floor number.
Persistent Changes:	This will allow building data from the Smart Bulbs to be categorized.
Related Requirements	

and Use Cases:	Building Schematics, Smart Bulb API Integration, Smart Device Registry
Test Cases:	1. Confirm that all desired devices have been registered. 2. Provide a Group Name (e.g. Hallway), and corresponding floor number and building ID. 3. Include all desired bulbs within defined group. 4. Verify that all bulbs and their corresponding IDs are linked with group name unique ID / table. 5. After a group is defined, verify that user can define more groups.

7.4 Register New User

Requirement Description:	Register as a new user on the website.
Rationale:	Making sure the user is able to register on the website with a unique ID.
Inputs:	First Name, Last Name, Username, email
Outputs:	Confirmation email should be sent out, user should be entered into user registry once confirmed, and be able to login to website.
Persistent Changes:	A new user will be entered into the database.
Related Requirements and Use Cases:	Fully Functioning Website, Database.
Initialization:	Go to website address and click *register* link.
Test Cases:	1. Enter First Name 2. Enter Last Name 3. Enter desired Username(Alphanumeric,No Special Characters) 4. Enter Email address 5. Click Register 6. Confirm Confirmation Email Link.

7.4.2 Login/Logout

Requirement Description:	User should be able to login to website with credentials and logout.
Rationale:	Making sure users are able to gain access to website using their credentials.
Inputs:	Username(Alphanumeric), Password(Case Sensitive)
Outputs:	User will gain access to website, and logout.
Persistent Changes	Users will gain access to website and Dashboard.
Related: Requirements and Use Cases:	User is registered in system.
Test Cases:	1. Enter Username 2. Enter Password 3. Click "Submit" 4. Click "Logout"

7.5 Recover Username/Password

Requirement Description:	Recover lost Username & Password
Rationale:	Make sure users can recover lost passwords or usernames.
Inputs:	Email used to register
Outputs:	Recovery email should be sent to the address.
Persistent Changes:	User will recover lost username and password.
Related Requirements and Use Cases:	Email provided is registered, Website is functional.
Test Cases:	1. Click "Forgot Username/Password" 2. Enter email address when prompted 3. Username and Password will be emailed to address.

7.6 Install Machine Learning Packages

Requirement Description:	Install Anaconda and machine learning scikit environment is properly set.
Rationale:	Make sure Python environment is ready for execution.
Inputs:	None
Outputs:	Python shell should open and show the installed version. Python commands should execute with no error.
Persistent Changes:	Machine Learning Scikit environment will be set.
Related Requirements and Use Cases:	Python is installed and updated.
Initialization:	Open Command Prompt on Windows Server and run as administrator
Test Cases:	1. Change directory to Python Home 2. Type python -V 3. Verify that output is : Python 3.5.2 :: Anaconda 4.2.0 (x86_64) 4. Type python.test 5. Command should execute with no error 6. Type python versions.py 7. Output should look this: scipy: 0.18.1 numpy: 1.11.1 matplotlib: 1.5.3 pandas: 0.18.1 statsmodels: 0.6.1 sklearn: 0.17.1

7.6.1 Test Machine Learning Functionalities

Requirement Description:	Provide a training file to an algorithm and test the accuracy of the prediction model
Rationale:	This will allow us to calibrate the prediction model and know how accurate it is based on output.
Inputs:	Training dataset
Outputs:	The algorithm should execute without errors and return a high accuracy.

Persistent Changes:	Accuracy of the machine learning algorithm will be known.
Related Requirements and Use Cases:	Machine Learning Environment setup with required libraries
Test Cases:	1. Ensure the training file is in the correct path directory 2. Execute the python mlib.py script 3. This should generate a prediction model. 4. Ensure that prediction model file is saved in the correct directory 5. Run cross validation tests on prediction model file 6. Check that the accuracy is within the acceptable threshold.

7.7 Smart Bulb API Integration

Requirement Description:	Connect API features and functions to be able to be used in correspondence to system database and website.
Rationale:	Once API features are connected, information will freeflow to our database.
Inputs:	None
Outputs:	Grouping, Registration of Smart Devices, Remote Toggle, and Remote Intensity Change functions are able to be used under website and functioning properly.
Persistent Changes:	API Data will be connected to the database and Tableau.
Related Requirements and Use Cases:	Philips Hue RESTful API Interface, Node.js application development, Link application to website
Initialization:	Open Existing BASH Shell or command prompt (e.g. Git Bash)
Test Cases:	1. Open command prompt and create a Node.js application using Express package. 2. Define and call API's functions for use within Node.js application. 3. After integration of API's basic functions, define user functions within Node.js application (e.g. remote toggle). 4. Verify that all functions are working properly by testing their corresponding tasks, and seeing if the outputs are the desired outputs.

5. Create user interface to link functions to usable buttons/sliders.
6. Verify that all buttons/sliders are working as intended.

7.7.1 Smart Bulb Toggle On / Off

Requirement Description:	Smart bulb(s) and clusters can remotely be toggled through developed application on website.
Rationale:	Ability to control Smart Bulb given to website manually.
Inputs:	Function call for Toggle
Outputs:	The bulbs should turn on or off, and should also turn on or off together if defined in a cluster.
Persistent Changes:	Users will gain the ability to control their Smart Bulbs via the website interface.
Related Requirements and Use Cases:	Smart Bulb API Integration
Initialization:	Press Toggle Button
Test Cases:	1. Press Toggle On/Off Button. 2. Verify if functionality works, if Light turns on, it works. If Light turns off, it works.

7.7.2 Smart Bulb Intensity Changer

Requirement Description:	Smart bulb(s) and clusters can remotely change brightness/intensity
Rationale:	In accordance with saving energy, brightness/intensity will be able to be controlled via website interface.
Inputs:	Function call for Intensity change
Outputs:	The bulbs should get dimmer or brighter depending on the value of brightness set.
Persistent Changes:	New Energy Use data will be fed to database.
Related Requirements	

and Use Cases:	Smart Bulb API Integration
Test Cases:	1. Use Intensity/Brightness Slider. 2. Verify if functionality works, if the brightness corresponds to the slider, it works.

7.7.3 Smart Bulb Wattage Output

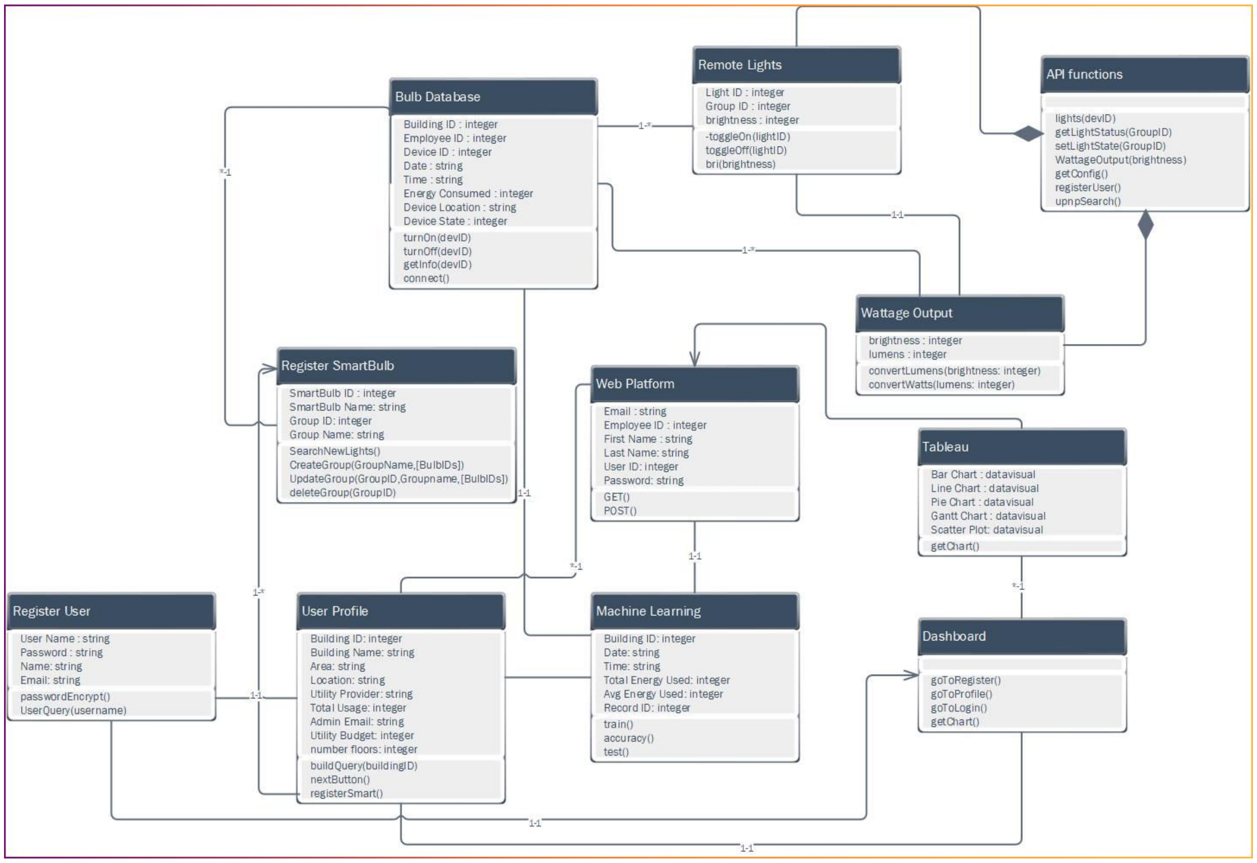
Requirement Description:	Smart bulb power usage per cluster/individual bulb is outputted and placed in corresponding table within database.
Rationale:	This is how data from the Smart Bulb will reach the database and improve the system.
Inputs:	Electricity
Outputs:	The wattage per second of bulbs/clusters is calculated through lumens per second which is calculated from value of brightness, and outputs into corresponding table within database.
Related Requirements and Use Cases:	Smart Bulb API Integration
Persistent Changes:	None.
Test Cases:	1. Start up API Connection with Website. 2. Set brightness at desired brightness, and manually calculate wattage per second through given formulas. 3. Compare value with value stored within database.

7.8 Integrate Data Visualizations

Requirement Description:	Integration of Data Visualization will allow easier to understand information and data analytics
Inputs:	Database data
Outputs:	Refined Tableau data visualization.
Persistent Changes:	None.
Related Requirements and Use Cases:	Functioning database and smart bulb API.
Test Cases:	1. Supply Tableau with the ability to recover data from the server. 2. Check to see if dashboard depicts correct graphs based off the data that will be supplied by the server.

Class Diagram

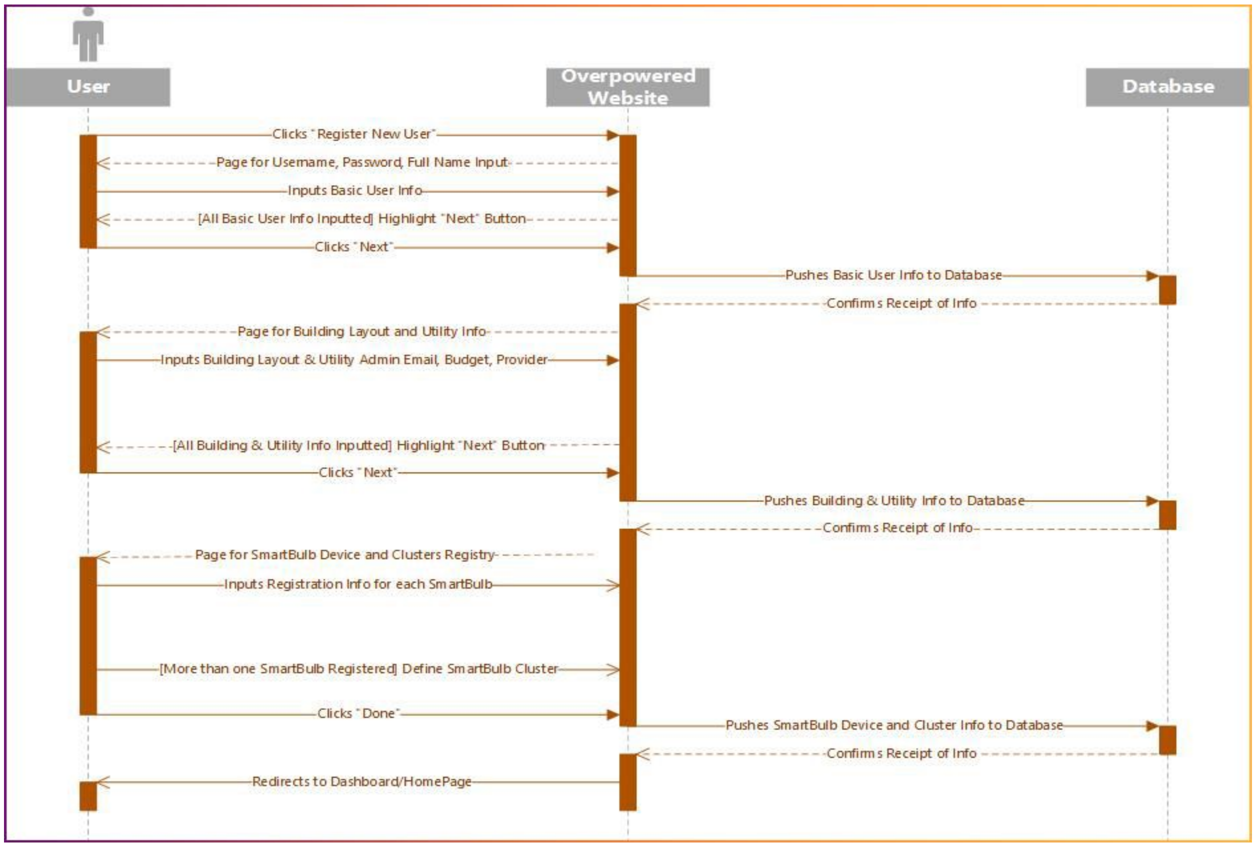
The Class Diagram file(s) can be referred to in Appendix 1.3.



Sequence Diagrams

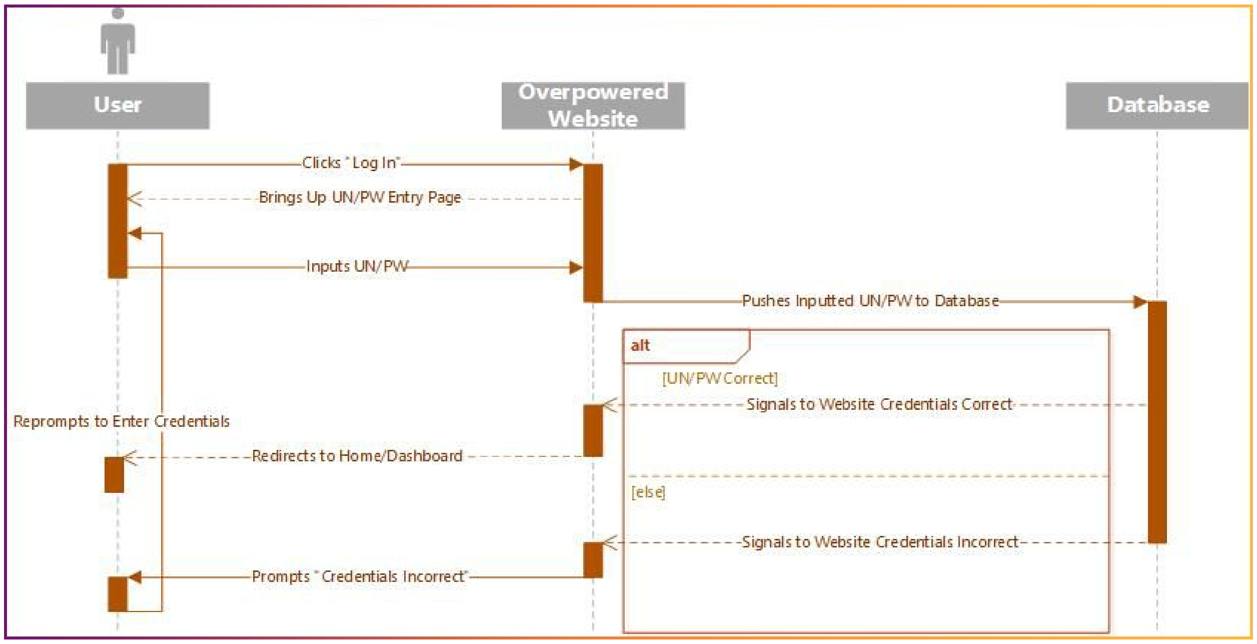
All Sequence Diagram files can be referred to in Appendix 1.4.

9.1 Register New User

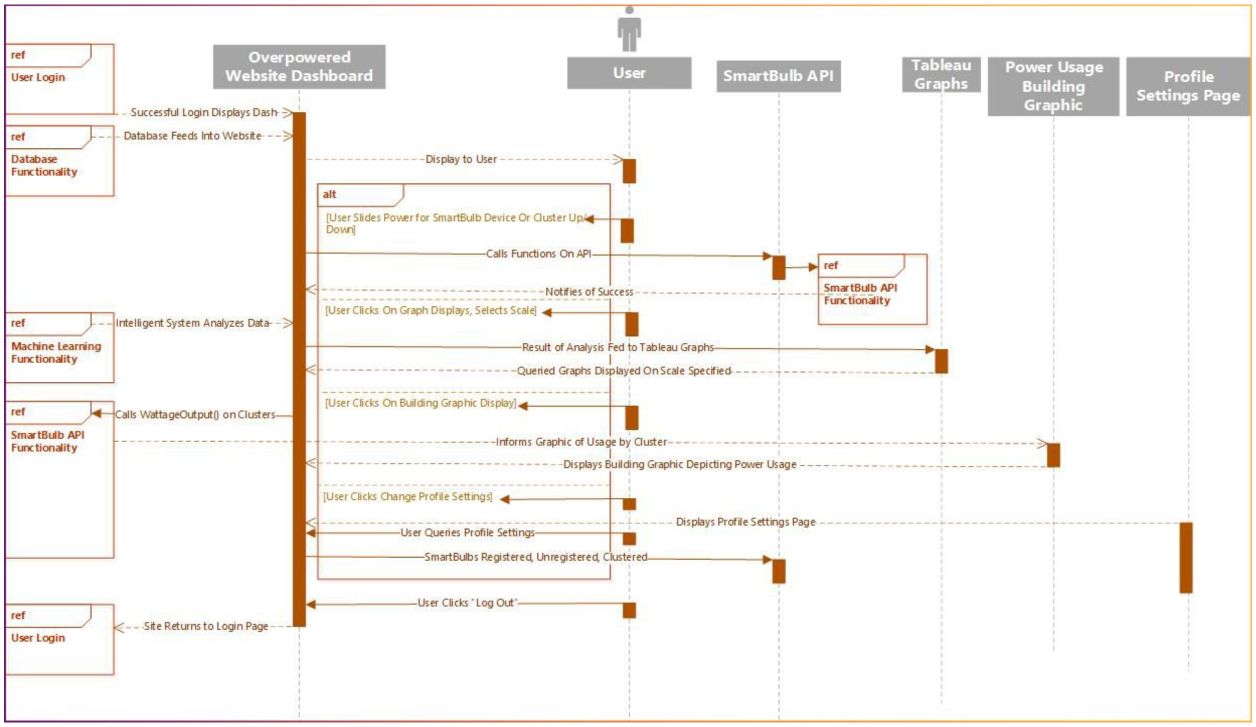


User
Overpowered Website
Database
Clicks "Register New User"
Page for Username, Password, Full Name Input
Inputs Basic User Info
[All Basic User Info Inputted] Highlight "Next" Button
Clicks "Next"
Pushes Basic User Info to Database
Confirms Receipt of Info
Page for Building Layout and Utility Info
Inputs Building Layout & Utility Admin Email, Budget, Provider
[All Building & Utility Info Inputted] Highlight "Next" Button
Clicks "Next"
Pushes Building & Utility Info to Database
Confirms Receipt of Info
Page for SmartBulb Device and Clusters Registry
Inputs Registration Info for each SmartBulb
[More than one SmartBulb Registered] Define SmartBulb Cluster
Clicks "Done"
Pushes SmartBulb Device and Cluster Info to Database
Confirms Receipt of Info
Redirects to Dashboard/HomePage

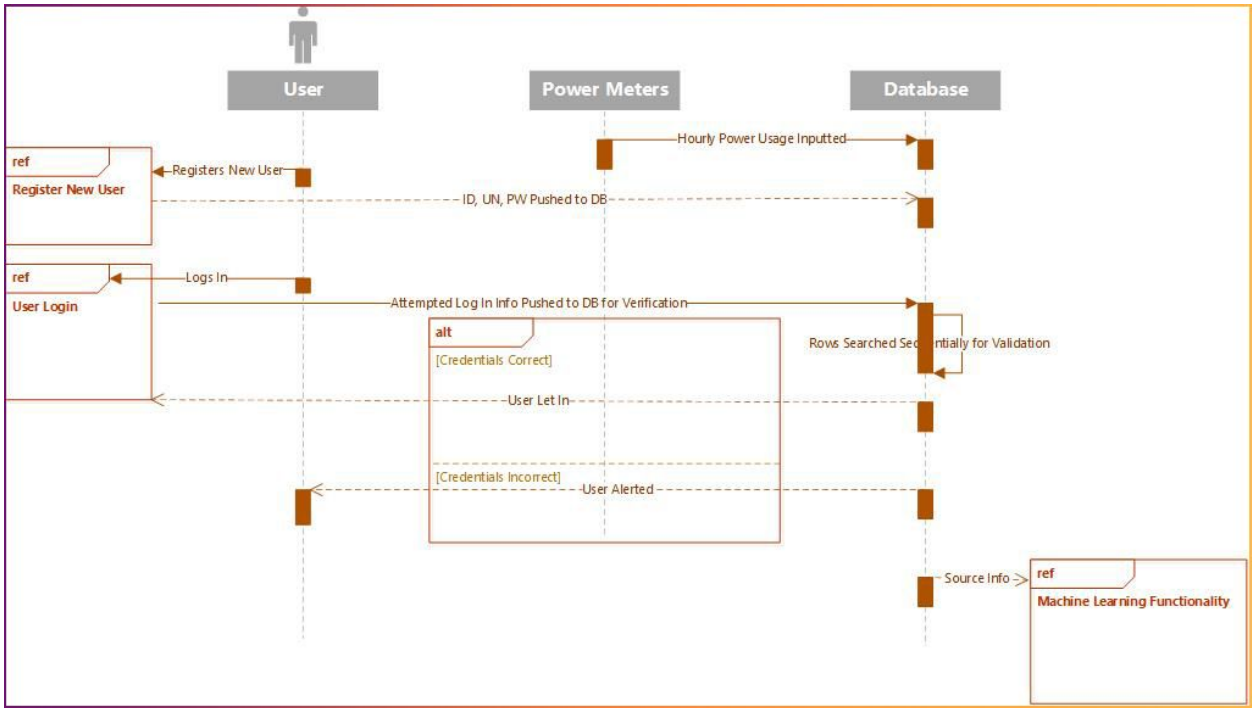
9.2 User Login


User
Overpowered Website
Database
Clicks "Log In"
Brings Up UN/PW Entry Page
Inputs UN/PW
Pushes Inputted UN/PW to Database
alt
[UN/PW Correct]
Signals to Website Credentials Correct
Reprompts to Enter Credentials
Redirects to Home/Dashboard
[else]
Signals to Website Credentials Incorrect
Prompts "Credentials Incorrect"

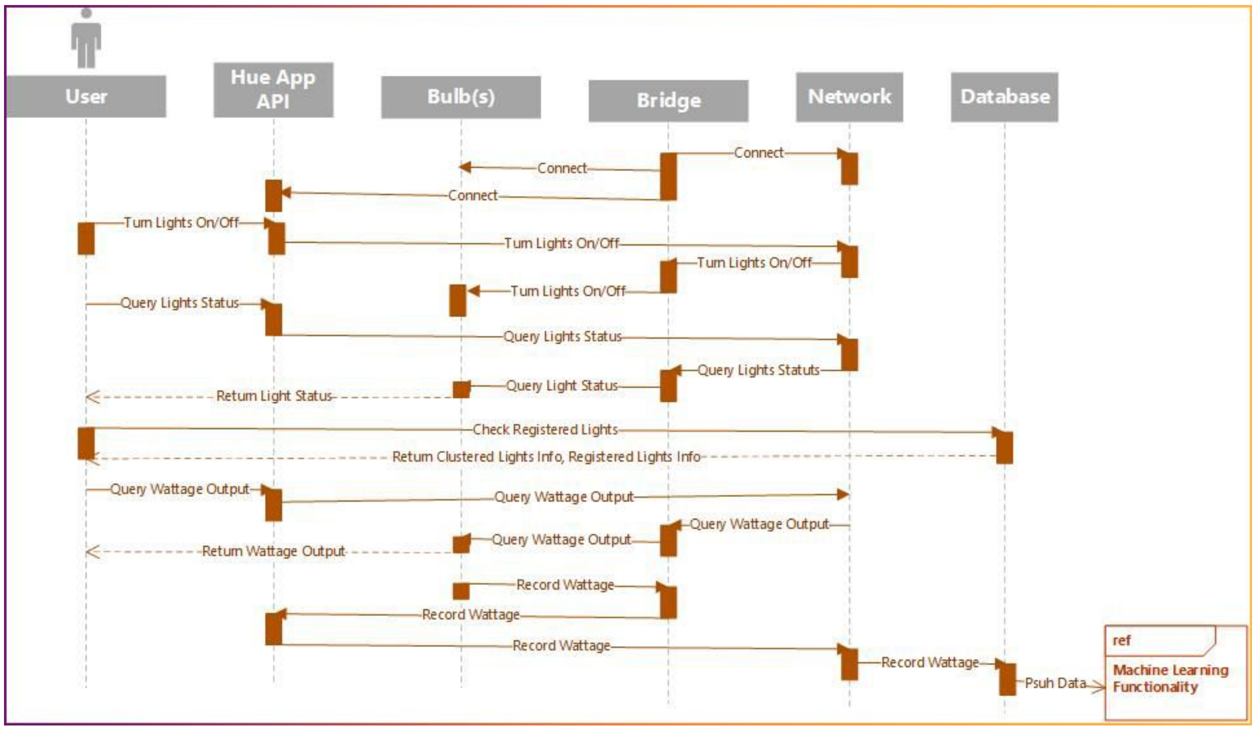
9.3 Dashboard Display Functionality


ref
User Login
Overpowered Website Dashboard
User
SmartBulb API
Tableau Graphs
Power Usage Building Graphic
Profile Settings Page
Successful Login Displays Dash
ref
Database Functionality
Database Feeds Into Website
Display to User
alt
[User Slides Power for SmartBulb Device Or Cluster Up/Down]
Calls Functions On API
ref
SmartBulb API Functionality
Notifies of Success
[User Clicks On Graph Displays, Selects Scale]
ref
Machine Learning Functionality
Intelligent System Analyzes Data
Result of Analysis Fed to Tableau Graphs
Queried Graphs Displayed On Scale Specified
[User Clicks On Building Graphic Display]
ref
SmartBulb API Functionality
Calls WattageOutput() on Clusters
Informs Graphic of Usage by Cluster
Displays Building Graphic Depicting Power Usage
[User Clicks Change Profile Settings]
Displays Profile Settings Page
User Queries Profile Settings
SmartBulbs Registered, Unregistered, Clustered
User Clicks "Log Out"
ref
User Login
Site Returns to Login Page

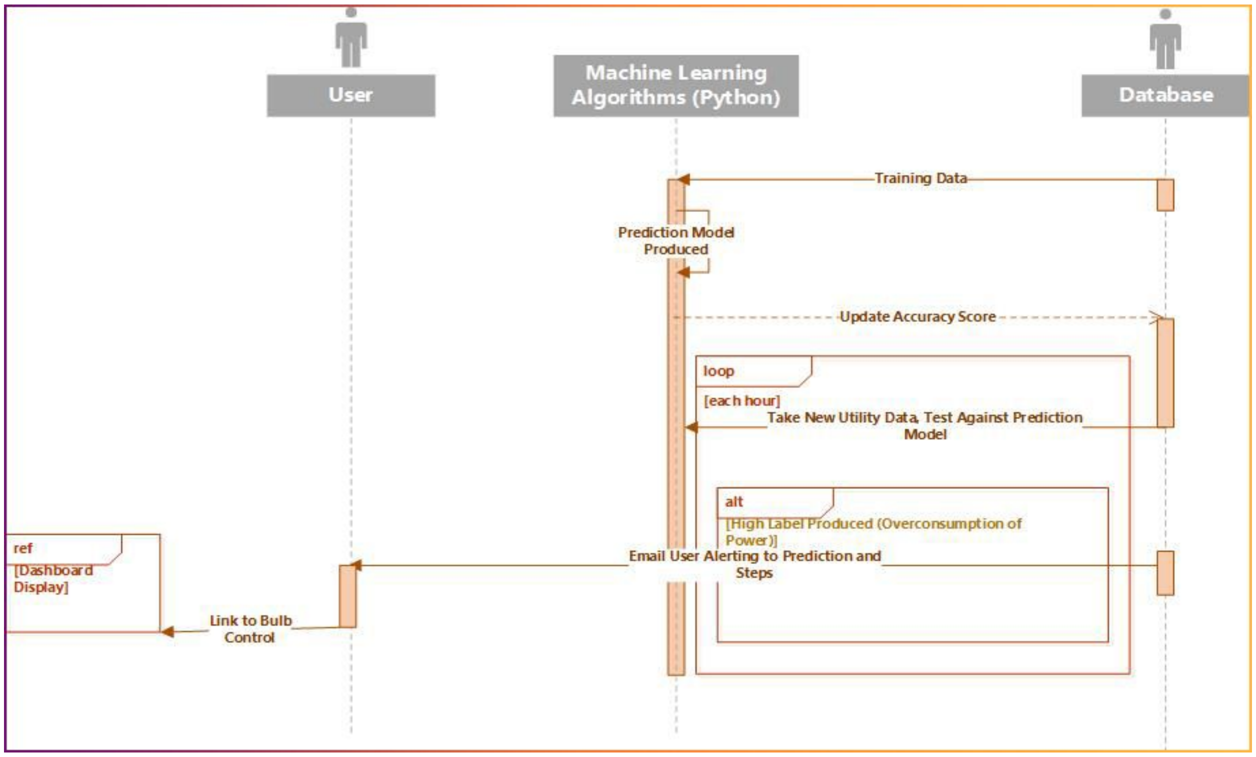
9.4 Database Functionality



9.5 Smart Bulb API Functionality



9.6 Machine Learning Functionality

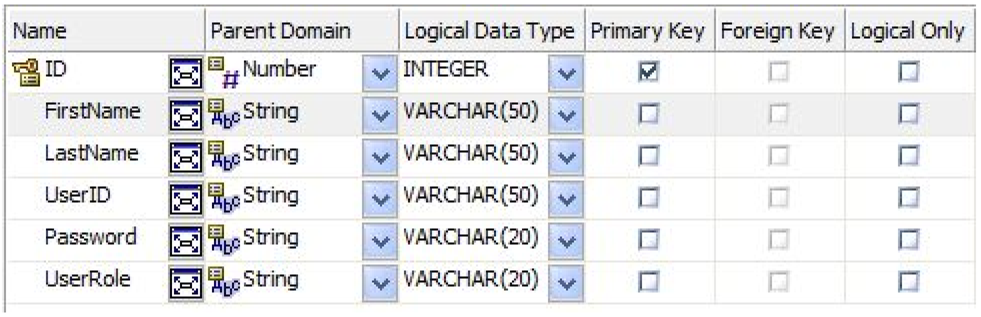

User
Machine Learning Algorithms (Python)
Database
Training Data
Prediction Model Produced
Update Accuracy Score
loop
[each hour]
Take New Utility Data, Test Against Prediction Model
alt
[High Label Produced (Overconsumption of Power)]
Email User Alerting to Prediction and Steps
ref
[Dashboard Display]
Link to Bulb Control

Database Specification and Analysis

Entity-Relationship Diagram follows the Database Specification and Analysis, and can also be referred to in Appendix 1.4. This project will also be using MySQL due to the security implementation, ease-of-access, and server configurations.

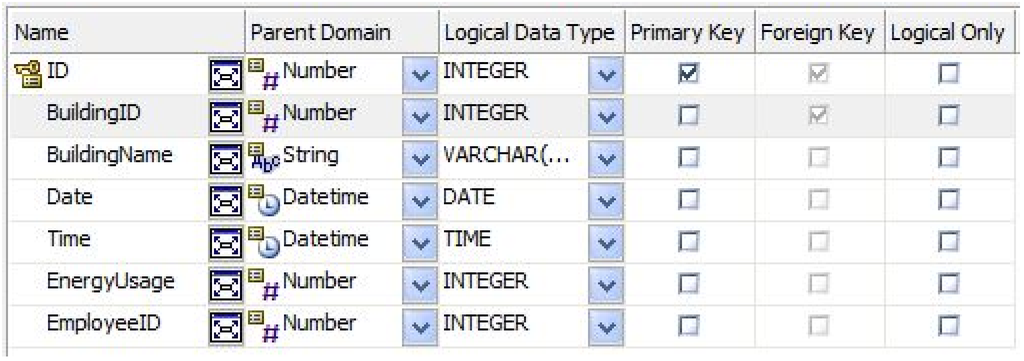
10.1 Entity Primary and Foreign Keys

10.1.1 Entity: Website



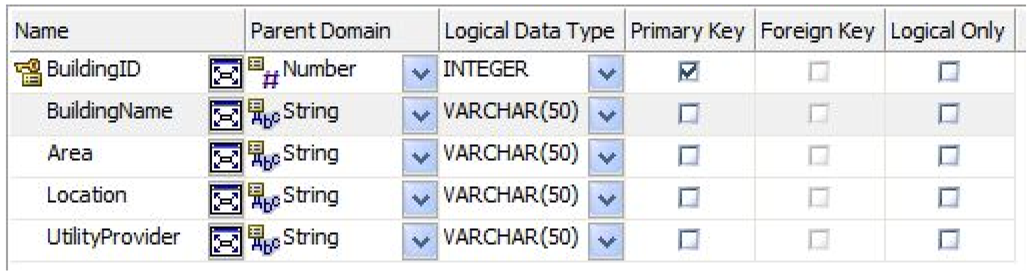
Name	Parent Domain	Logical Data Type	Primary Key	Foreign Key	Logical Only
ID	Number	INTEGER	☑	☐	☐
FirstName	String	VARCHAR(50)	☐	☐	☐
LastName	String	VARCHAR(50)	☐	☐	☐
UserID	String	VARCHAR(50)	☐	☐	☐
Password	String	VARCHAR(20)	☐	☐	☐
UserRole	String	VARCHAR(20)	☐	☐	☐

10.1.2 Entity: Tableau



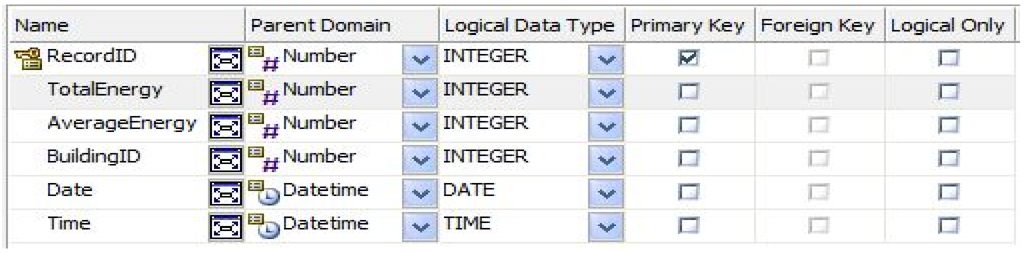
Name	Parent Domain	Logical Data Type	Primary Key	Foreign Key	Logical Only
ID	Number	INTEGER	✓	✓	☐
BuildingID	Number	INTEGER	☐	✓	☐
BuildingName	String	VARCHAR(...	☐	☐	☐
Date	Datetime	DATE	☐	☐	☐
Time	Datetime	TIME	☐	☐	☐
EnergyUsage	Number	INTEGER	☐	☐	☐
EmployeeID	Number	INTEGER	☐	☐	☐

10.1.3 Entity: ClientInfo



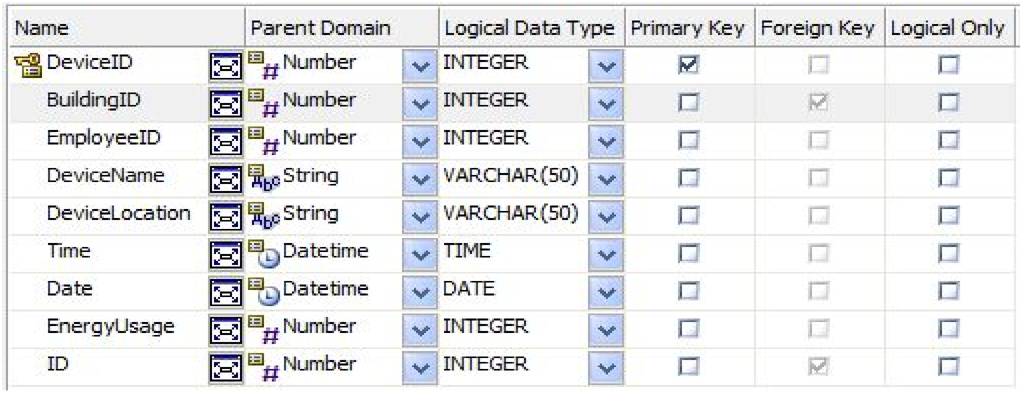
Name	Parent Domain	Logical Data Type	Primary Key	Foreign Key	Logical Only
BuildingID	Number	INTEGER	✓	☐	☐
BuildingName	String	VARCHAR(50)	☐	☐	☐
Area	String	VARCHAR(50)	☐	☐	☐
Location	String	VARCHAR(50)	☐	☐	☐
UtilityProvider	String	VARCHAR(50)	☐	☐	☐

10.1.4 Entity: MachineLearning



Name	Parent Domain	Logical Data Type	Primary Key	Foreign Key	Logical Only
RecordID	Number	INTEGER	✓	☐	☐
TotalEnergy	Number	INTEGER	☐	☐	☐
AverageEnergy	Number	INTEGER	☐	☐	☐
BuildingID	Number	INTEGER	☐	☐	☐
Date	Datetime	DATE	☐	☐	☐
Time	Datetime	TIME	☐	☐	☐

10.1.5 Entity: SmartDevice

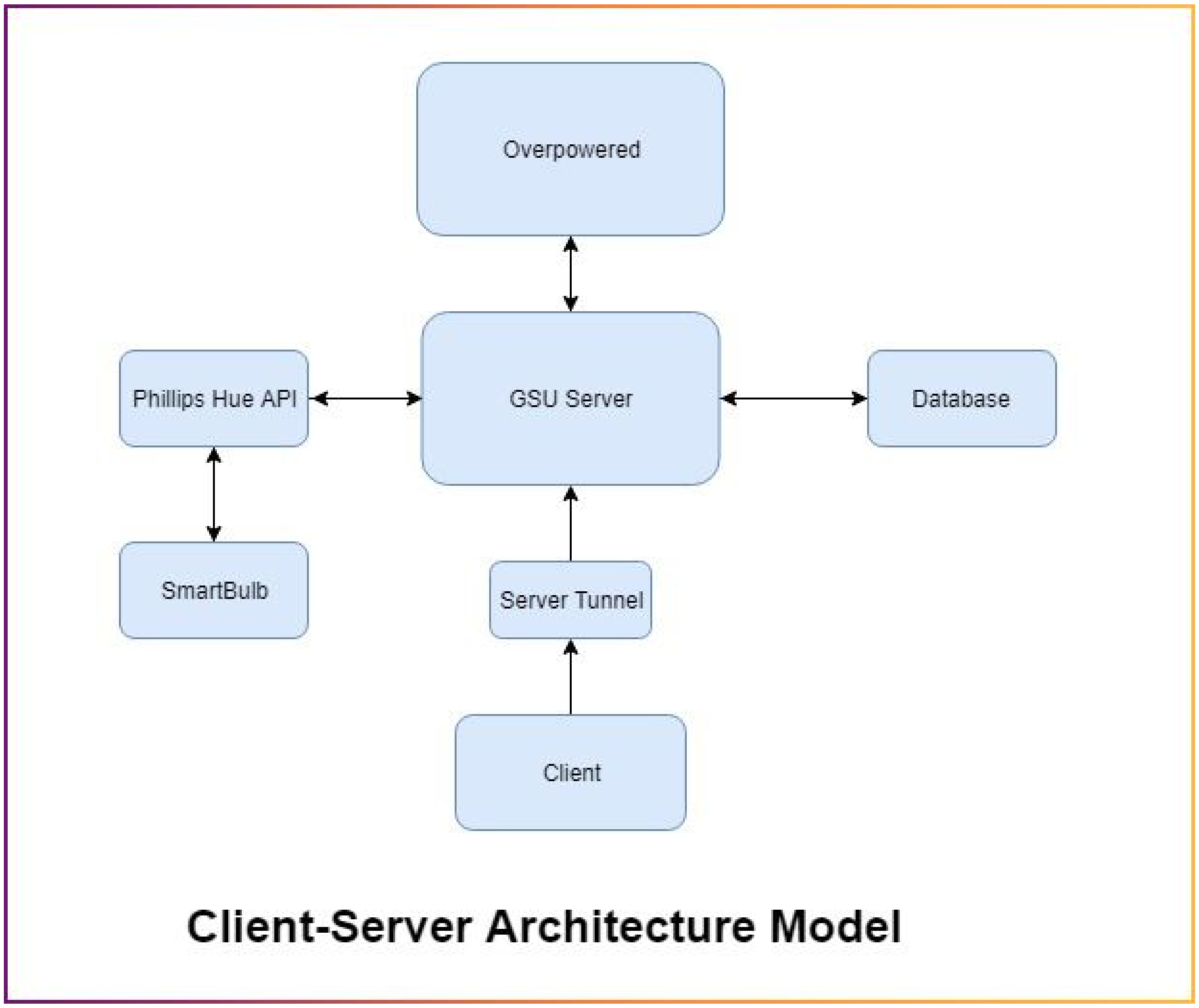


Name	Parent Domain	Logical Data Type	Primary Key	Foreign Key	Logical Only
DeviceID	Number	INTEGER	✓	☐	☐
BuildingID	Number	INTEGER	☐	✓	☐
EmployeeID	Number	INTEGER	☐	☐	☐
DeviceName	String	VARCHAR(50)	☐	☐	☐
DeviceLocation	String	VARCHAR(50)	☐	☐	☐
Time	Datetime	TIME	☐	☐	☐
Date	Datetime	DATE	☐	☐	☐
EnergyUsage	Number	INTEGER	☐	☐	☐
ID	Number	INTEGER	☐	✓	☐

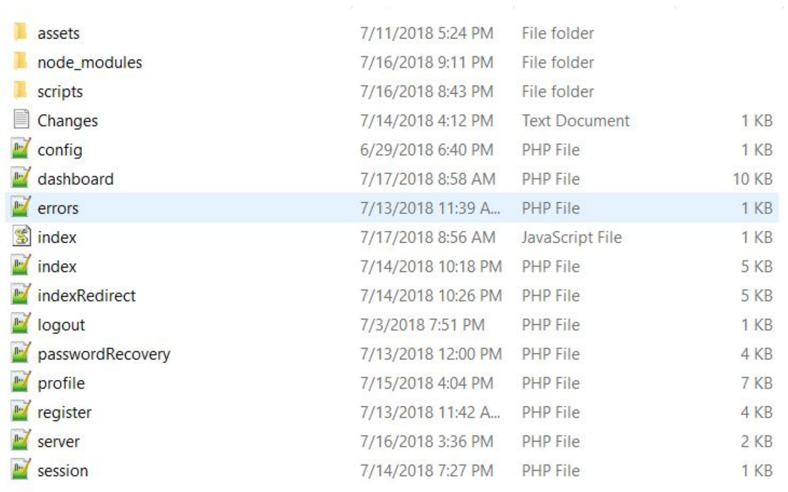
10.2 Design Pattern Specification and Measures

Our architectural design pattern will be client-server, and our Class Diagram has Nominal Coupling and High Cohesion. There are a couple of classes that are dependent on one another, hence the nominal coupling, and has high cohesion due to the fact that each class mainly focuses on their specific methods. Only a few methods are shared between classes.

10.2.1 Client-Server Architecture Model



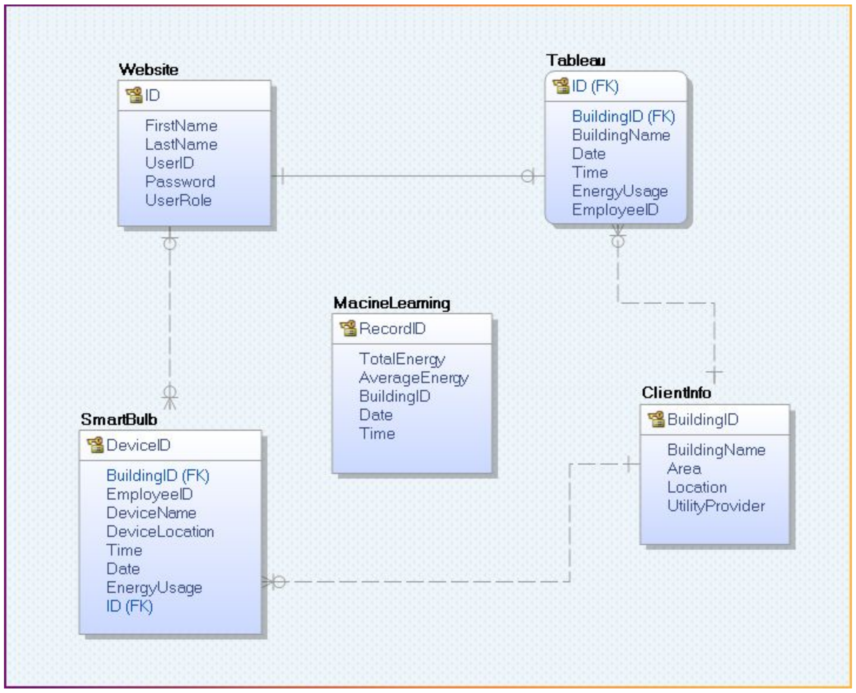
10.2.2 Project Structure



assets	7/11/2018 5:24 PM	File folder	
node_modules	7/16/2018 9:11 PM	File folder	
scripts	7/16/2018 8:43 PM	File folder	
Changes	7/14/2018 4:12 PM	Text Document	1 KB
config	6/29/2018 6:40 PM	PHP File	1 KB
dashboard	7/17/2018 8:58 AM	PHP File	10 KB
errors	7/13/2018 11:39 A...	PHP File	1 KB
index	7/17/2018 8:56 AM	JavaScript File	1 KB
index	7/14/2018 10:18 PM	PHP File	5 KB
indexRedirect	7/14/2018 10:26 PM	PHP File	5 KB
logout	7/3/2018 7:51 PM	PHP File	1 KB
passwordRecovery	7/13/2018 12:00 PM	PHP File	4 KB
profile	7/15/2018 4:04 PM	PHP File	7 KB
register	7/13/2018 11:42 A...	PHP File	4 KB
server	7/16/2018 3:36 PM	PHP File	2 KB
session	7/14/2018 7:27 PM	PHP File	1 KB

10.3 Entity-Relationship Diagram

The Entity-Relationship Diagram file can be referred to in Appendix 1.5.



Implementation and Deployment

All documentation and code is linked to our GitHub provided in Appendix 1.0, or found in their corresponding Appendices.

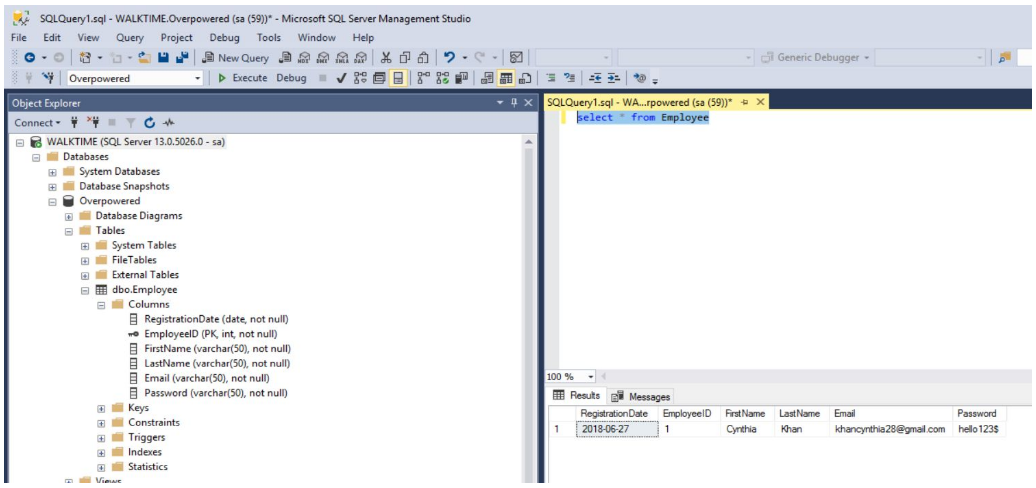
11.1 User ReadMe's

11.1.0 Login to Remote Desktop and Website

To connect to website:

1. Connect to walktime server via Remote Desktop. This requires either a VPN (secureaccess.gsu.edu) or GSU WiFi connection.
    a. To access our walktime server via Remote Desktop, credentials of one of our WalktimeError members will be required:
        i. UN: rle5@student.gsu.edu
        ii. PW: G14554n1m415
2. Run WAMP on server with administrative privileges (easily access this by typing 'WAMP' into the Windows search bar).
3. Run ngrok as administrator (within the C drive, there will be an associated folder with the ngrok application and a ReadMe describing the below instructions).
4. A. Type 'ngrok http -subdomain=walktime 80' into ngrok command line.
    a. The website will now be accessible via walktime.ngrok.io
5. The address will now be accessible to anyone connected to the GSU network for the next 8 hours.

11.1.1 Database Table Implementation


SQLQuery1.sql - WALKTIME.Overpowered (sa (59))* - Microsoft SQL Server Management Studio
File Edit View Query Project Debug Tools Window Help
New Query
Generic Debugger
Overpowered
Execute Debug
Object Explorer
Connect
SQLQuery1.sql - WA...rpowered (sa (59))*
100 %
Results
Messages

```
WALKTIME (SQL Server 13.0.5026.0 - sa)
  Databases
    System Databases
    Database Snapshots
    Overpowered
      Database Diagrams
      Tables
        System Tables
        FileTables
        External Tables
        dbo.Employee
          Columns
            RegistrationDate (date, not null)
            EmployeeID (PK, int, not null)
            FirstName (varchar(50), not null)
            LastName (varchar(50), not null)
            Email (varchar(50), not null)
            Password (varchar(50), not null)
          Keys
          Constraints
          Triggers
          Indexes
          Statistics
      Views
```

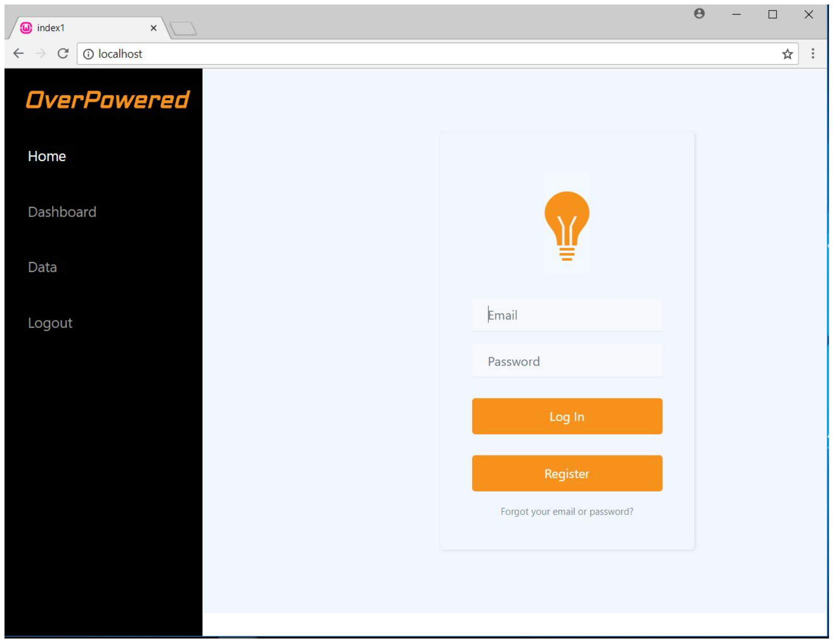
```sql
select * from Employee
```

	RegistrationDate	EmployeeID	FirstName	LastName	Email	Password
1	2018-06-27	1	Cynthia	Khan	khancynthia28@gmail.com	hello123$

11.1.1.1 Database Table ReadMe

Since this is the backend, the user will not interact with the MySQL tables directly. Information here is added and removed upon user registration and it is called upon using PHP when the user logs in.

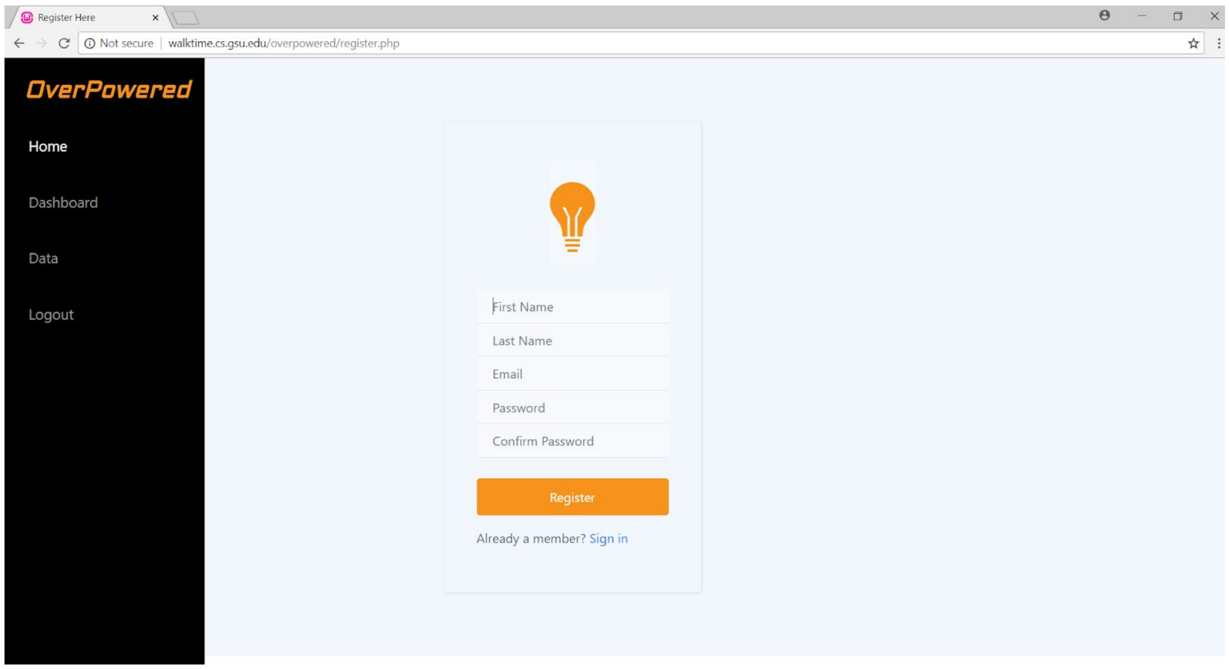
11.1.2 User Login Implementation



11.1.2.1 User Login ReadMe

1. Ensure you are connected to the GSU WiFi network.
2. Go to walktime.us or walktime.ngrok.io.
3. Recall your login information. If this is your first time logging in, click "Register" and refer to section 15.3.1.
4. Input the email you used to login as well as the password you used to login into the appropriately labeled boxes.
    a. For Dr. Awad Mussa, your login information is:
        i. Email: " amussa@gsu.edu "
        ii. Password: " youShouldUseAgile "
5. Click "Log In".

11.1.3 Register User Implementation

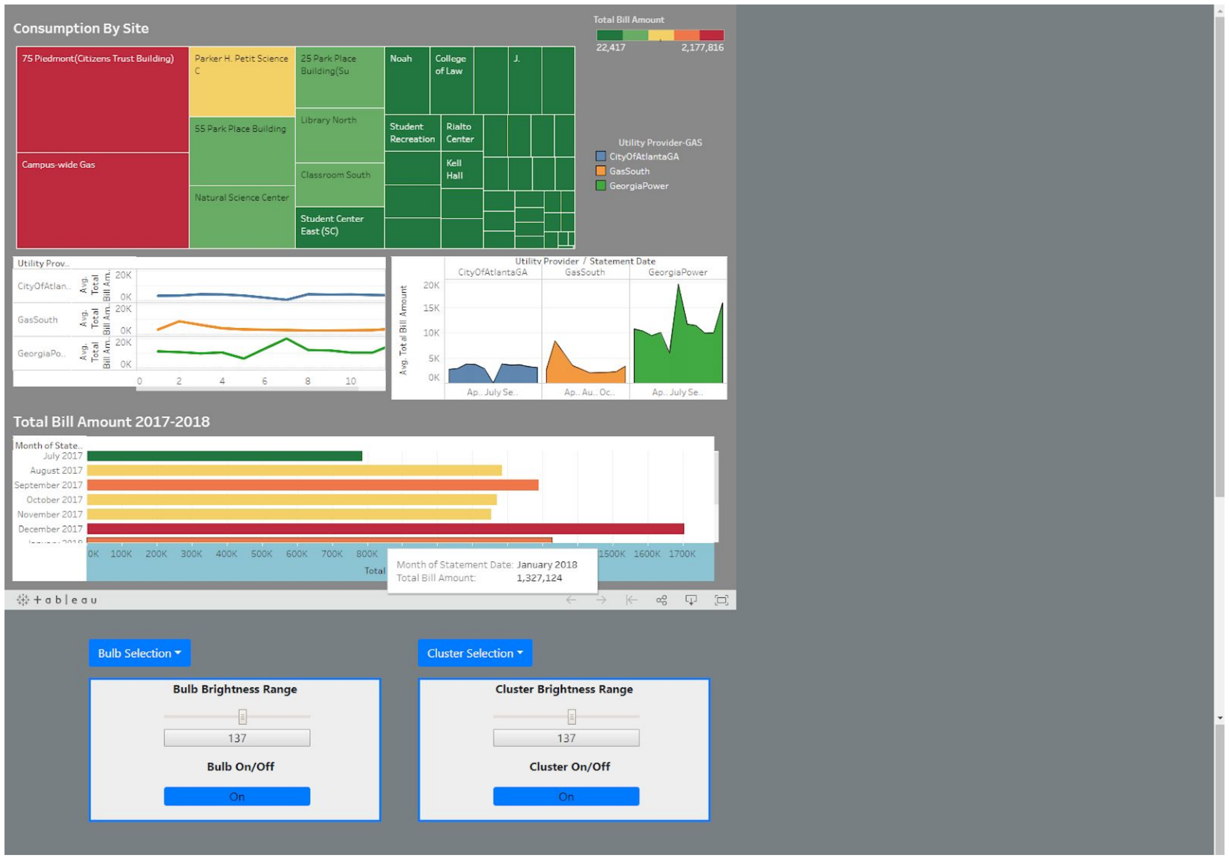


15.1.3.1 Register User ReadMe

1. Ensure you are connected to the GSU WiFi network.
2. Go to walktime.cs.gsu.edu/overpowered/index.php
3. If you already have an account with login info, refer to section 15.2.1.
4. Click "Register".
5. Input your First Name, Last Name, email, and desired password, and then retype your desired password precisely into the appropriately labeled sections.

6. Click “Register”.
7. If you didn’t care to listen to item 3, you can click “Sign in” below the “Register” button.

11.1.4 Dashboard Display Functionality



11.1.4.1 Dashboard Functionality ReadMe

1. Once logged in (see 15.2) user should be taken to Dashboard page.
2. From here, the user may interface with the Smartbulb controls.
   a. They may select a specific labeled cluster of bulbs via a drop down menu..
      i. They may increase/decrease the brightness from 0-244.
      ii. They may turn the clusters on and off.
   b. They may select individual bulbs via a dropdown menu.
      i. They may increase/decrease the brightness from 0-244.
      ii. They may turn the clusters on and off.
3. The user may also access varying display info.
   a. Clicking Bar Graph displays a bar graph view by bulb cluster.

b. Clicking line Graph shows power consumption overall over varying time scales selectable by the user.
c. Clicking Pie Chart depicts the proportion of overall power usage that each cluster has used.

11.1.5 Logout ReadMe

1. Should a user be finished interfacing with the website, they may click "logout" to be redirected to the Login screen.

11.1.6 Update Password ReadMe

1. While logged in, click "Profile" section in the sidebar menu.
2. Type in new password you would wish to use
3. Confirm new password in box below
4. Click "Update" and password will be changed.
5. If passwords match, that will be the new password used to login for that user.

11.1.6.2 Update Email ReadMe

1. While logged in, click "Profile" section in the sidebar menu.
2. Type in new email address you would wish to use as a login.
3. Confirm new email address in the box below.
4. Click blue "Update" button in the box below
5. If email's match, that will be the new email used to sign in for that user.

11.1.7 Define/Add Bulb Cluster ReadMe

1. While logged in, click "Profile" section in the sidebar menu.
2. Clicking on the "Bulb Clusters" button will open a dropdown menu which displays current clusters you have.
3. To define new cluster, type in the desired cluster name below the "Bulb Clusters" button and click "Submit".
4. New cluster will then be part of "Bulb Clusters" drop down menu.

11.1.7.1 Define/Add SmartBulb ReadMe

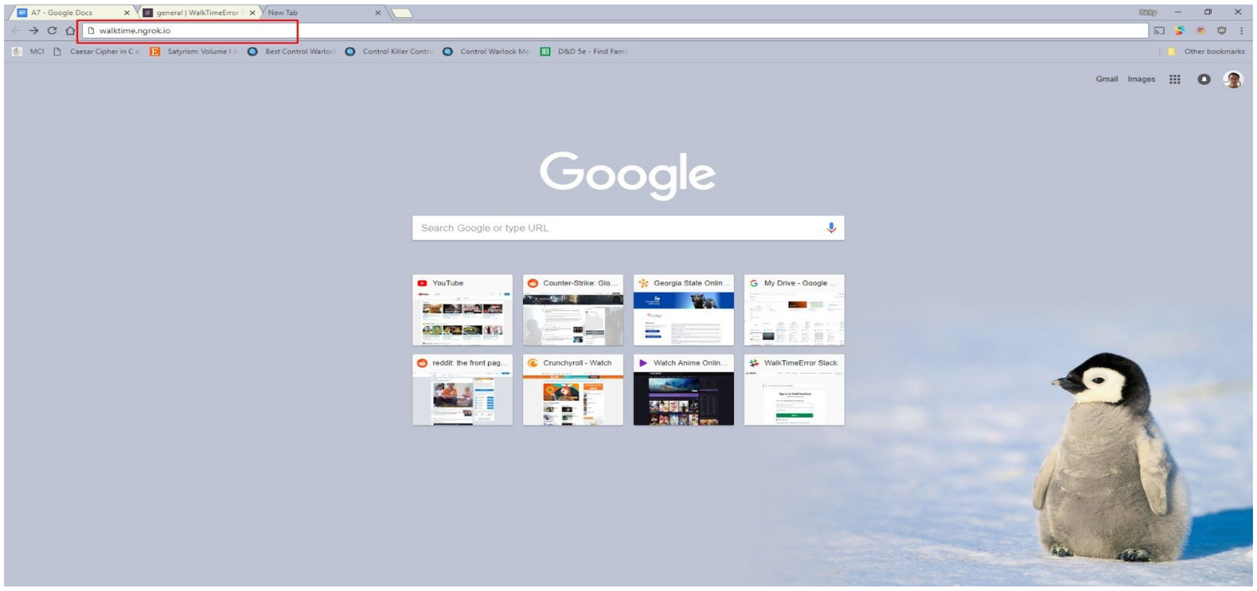
1. While logged in, click "Profile" section in the sidebar menu.
2. Click on "Bulb Clusters" to select a certain 'group' of bulbs to work with.
3. Clicking on the "Bulb ID" button will open a dropdown menu which displays current bulbs you have in the selected cluster.
4. Type in the SmartBulb ID number included with your Philips Smart Bulb, and click "Submit"

5. Smart bulb will be added to the selected cluster.

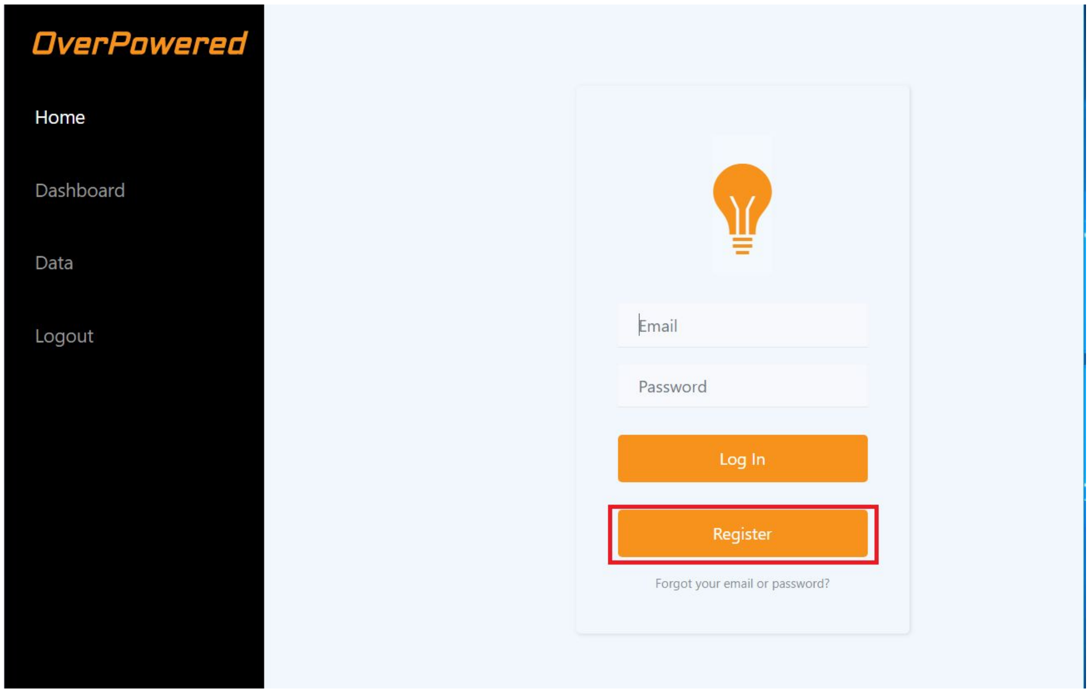
11.2 How To Guide; No Installation Required!

0. Ensure that the tunnel and local server (WAMP) are running on the walktime server. Email nyasarturk1@student.gsu.edu if this is not the case.

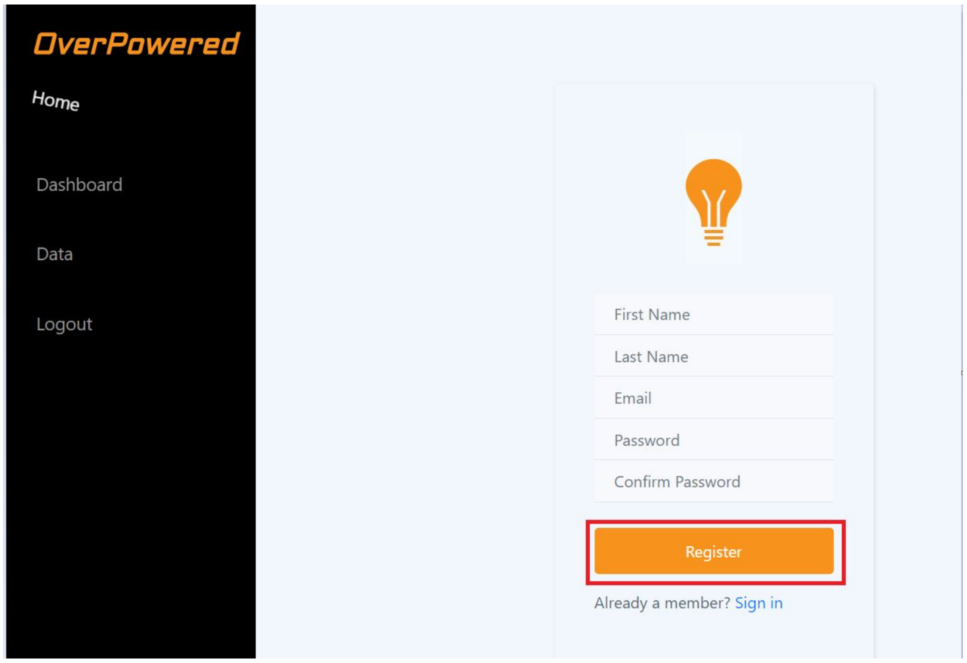
1.Open web browser of choice and type in https://walktime.ngrok.io/ into the address bar. Press Enter.



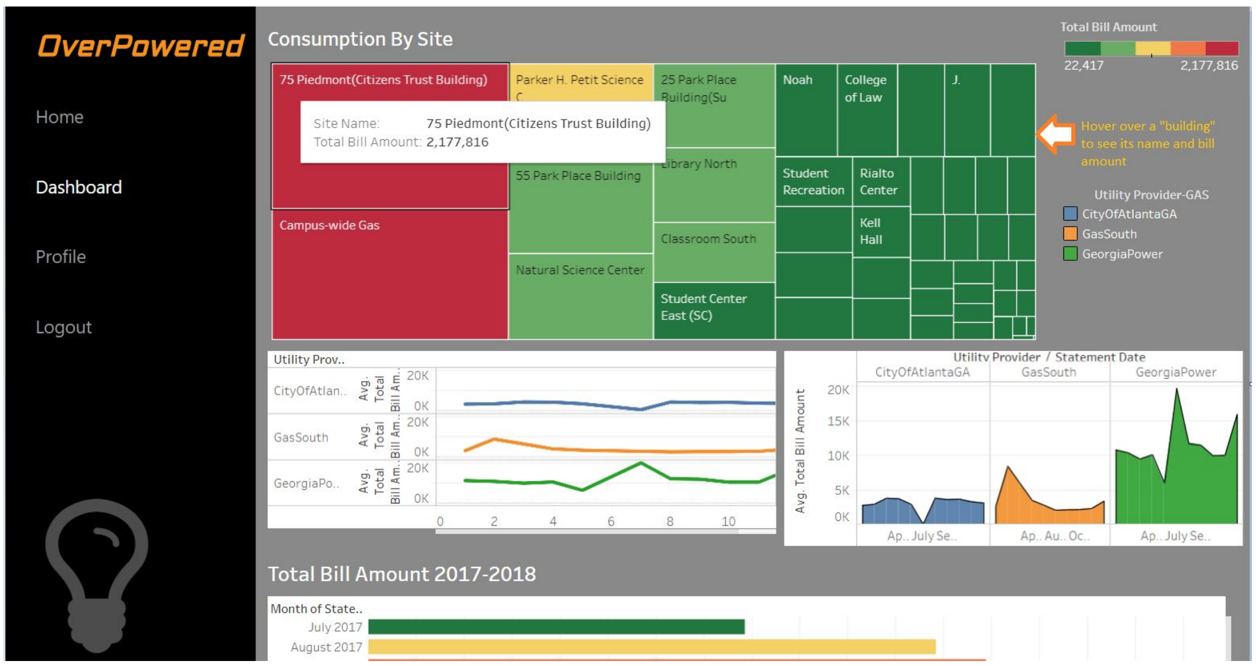
2. Click Register User to register new account.

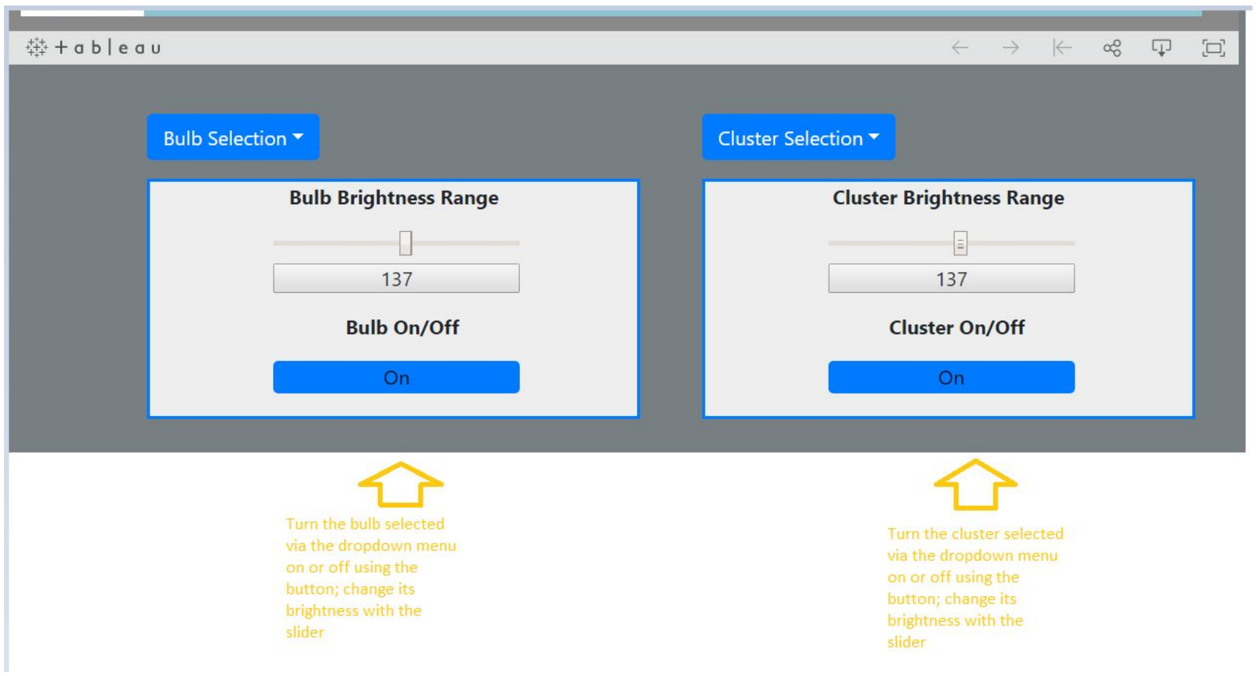


3. Fill in required information and click the Register Button, you will be redirected to the Login Page.



4. Type in the information provided when making the account, and click the Login Button. You will be redirected to the website's Dashboard Page. From here, you can access all of the website's functions by clicking on any of the words on the sidebar panel. You can edit profile information, look at the dashboard, remotely toggle bulbs and clusters, as well as change their brightness.




+ a b | e a u
Bulb Selection
Bulb Brightness Range
137
Bulb On/Off
On
Cluster Selection
Cluster Brightness Range
137
Cluster On/Off
On
Turn the bulb selected via the dropdown menu on or off using the button; change its brightness with the slider
Turn the cluster selected via the dropdown menu on or off using the button; change its brightness with the slider

5. If the user needs to register a new bulb or change utility info, they can navigate to the Profile page to use these features.

   a. Changing bulb info:

      i. The "Cluster Name" input box corresponds to the name of the room or light set you want to group together. Bulbs must be registered prior to grouping into clusters.

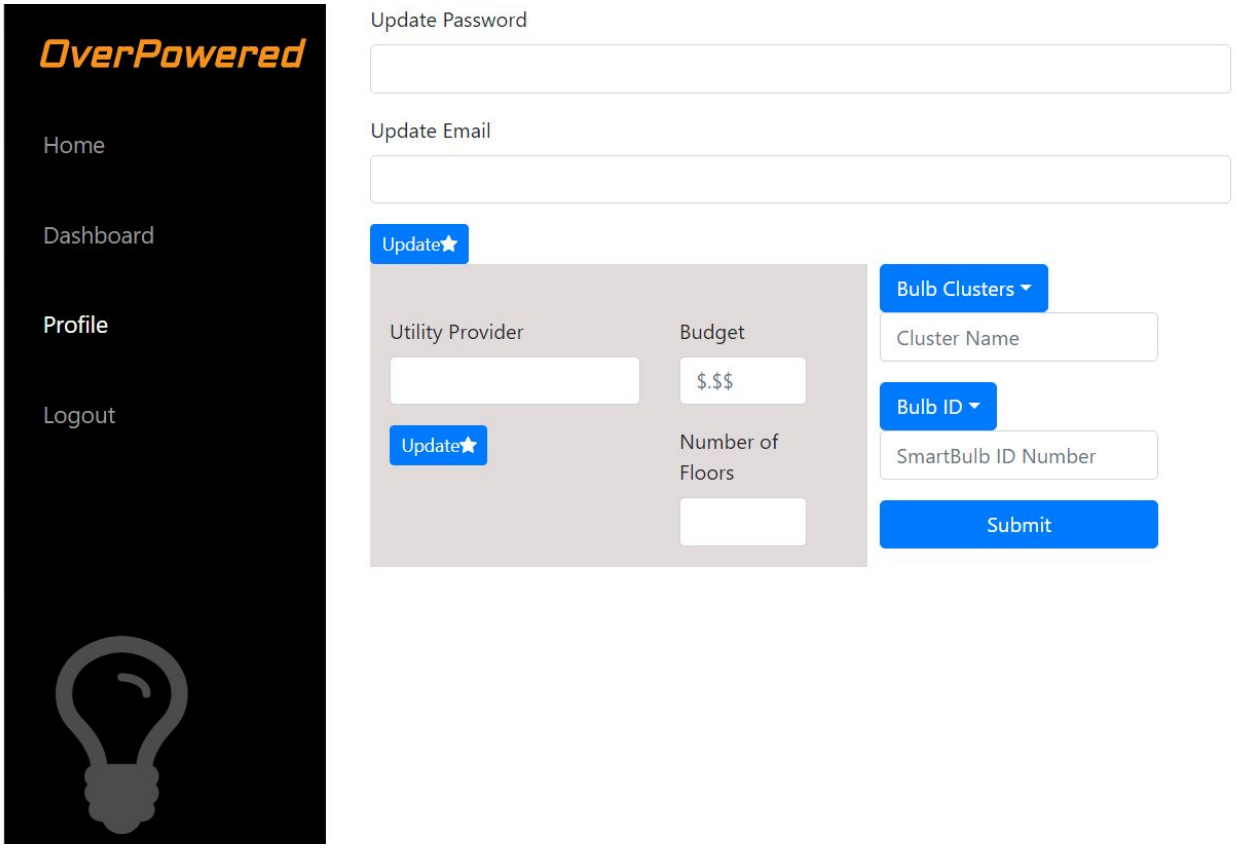
      ii. The "SmartBulb ID number" input box corresponds to the physical bulb idea found on your Philips Hue SmartBulb near the base in black text next to "Serial

No..”. Only input the first six characters. This is how the user registers a new SmartBulb.

iii. Click the large “Submit” button to change/input the typed bulb information above it.

b. The grayed box corresponds to utility information specified within the input boxes. Press the “Update” button within it once proper information has been inputted if the user desires to change their utility info.

c. The Update Password and Update Email input boxes can be provided with a new email address or password and submitted via the below “Update” button that is not within the grayed section.



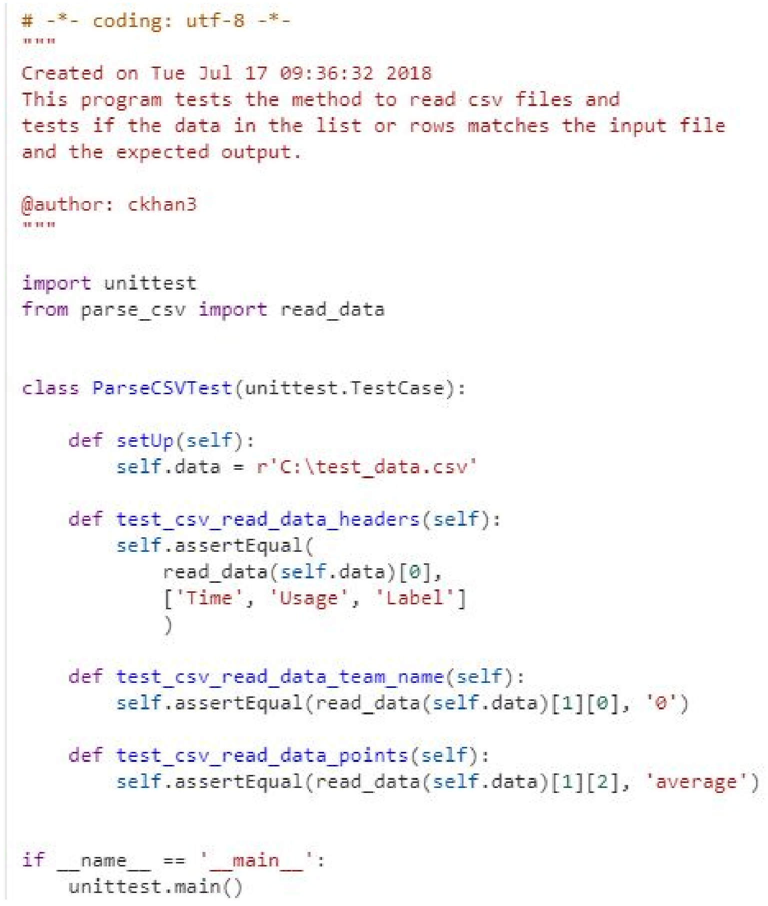
6. To log out, click the “Logout” tab along the left of the OverPowered website.

Testing

The following section contains Black Box Testing as well as White Box Testing during the Testing Phase of our Project.

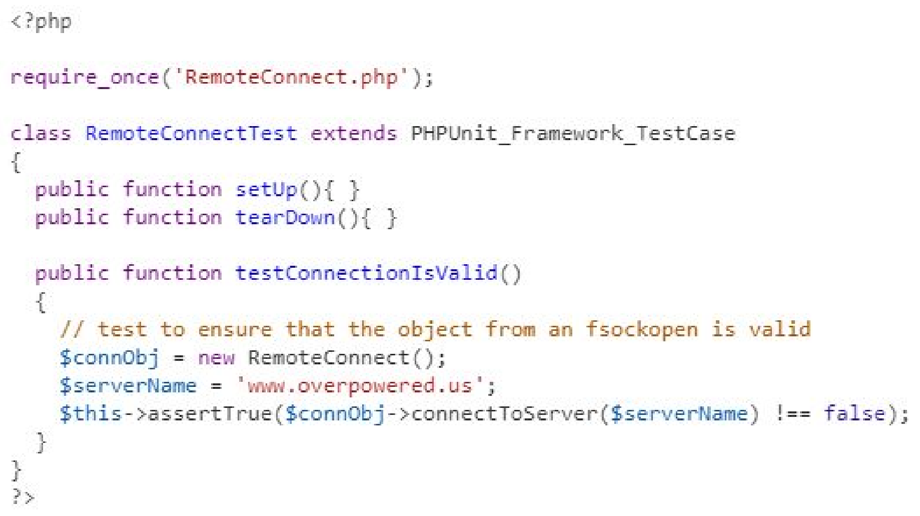
12.1 WhiteBox Testing

12.1.1 Python Test



```python
# -*- coding: utf-8 -*-
"""
Created on Tue Jul 17 09:36:32 2018
This program tests the method to read csv files and
tests if the data in the list or rows matches the input file
and the expected output.

@author: ckhan3
"""

import unittest
from parse_csv import read_data


class ParseCSVTest(unittest.TestCase):

    def setUp(self):
        self.data = r'C:\test_data.csv'

    def test_csv_read_data_headers(self):
        self.assertEqual(
            read_data(self.data)[0],
            ['Time', 'Usage', 'Label']
            )

    def test_csv_read_data_team_name(self):
        self.assertEqual(read_data(self.data)[1][0], '0')

    def test_csv_read_data_points(self):
        self.assertEqual(read_data(self.data)[1][2], 'average')


if __name__ == '__main__':
    unittest.main()
```

12.1.2 PHP Test



```
<?php

require_once('RemoteConnect.php');

class RemoteConnectTest extends PHPUnit_Framework_TestCase
{
  public function setUp(){ }
  public function tearDown(){ }

  public function testConnectionIsValid()
  {
    // test to ensure that the object from an fsockopen is valid
    $connObj = new RemoteConnect();
    $serverName = 'www.overpowered.us';
    $this->assertTrue($connObj->connectToServer($serverName) !== false);
  }
}
?>
```

12.2 BlackBox Testing

Test Case ID	Test Cases	Pre-condition	Setup	Input	Expected Output
1	Test if user is able to login successfully.	User must be registered	1)Enter input(correct)username and password on the respective fields 2)click submit/login	correct username,correct password	User must successfully login to the web page
2	Test with valid username and empty password such that login must get failed	User must be registered already	1)enter the valid username in the user id and enter no password in the password field	valid username and empty password	Proper error must be displayed and prompt to enter login again
3	Test with empty username and valid password such that login must get failed	User must be registered already	1)leave the username empty in the user id and enter a valid	empty username and valid password	Proper error must be displayed and prompt to enter login

			user's password in the password field		again
4	Test with empty username and/or empty password and check if login fails	Index page should be open, empty form	1)Leave form empty/or one input empty 2)Attempt to log in	1)Enter nothing in the mail id and password field 2)click submit button	1)Enter nothing in the mail id and password field 2)click submit button
5	Check of the password is masked on the screen i.e., password must be in bullets or asterisks	Form should be filled	1) Enter the password field with some characters	some password(can be a Registered/un registered)	The password field should display the characters in asterisks or bullets such that the password is not visible on the screen
6	Test if unregistered users are able to login to the site	Must be logged out	1)Enter input(incorrect)username and password on the respective fields 2)click submit/login	Incorrect username and/or incorrect password	Proper error must be displayed and prompt to enter login again
7	Test if Register button takes to registration page	Current page must be on index page	1)Click on Register button	Register click	User must be directed to registration form page
8	Test if Registration form enforces required fields	Current page must be registration page	1)Leave a few fields blank 2)Fill the rest of the fields 3)Click submit	Empty fields	Proper error must be displayed and prompt user to enter all required information
9	Test if submitting registration	User must not have prior	1)Fill out all required fields	New user information	Proper success

	correctly registers the new user	registration	2)Click submit		message must be displayed notifying that the user is registered
10	Check if selecting back button (after logging out) will cause user to come back to being logged in.	User must have been logged in, and clicked logout	1)Login with registered username and password 2)once you are logged in, sign out of the site 3)press back button	Registered username and password	User shouldn't be signed in to his account rather a general web page must be visible
11	Log-out of the site when pressing backpage button	User must be registered already, and logged on.	1) Login to the site using registered username and password 2)now press backspace	Back-page button.	User must log-out of the site properly
12	Test that all links are active and directing to the proper pages	User must be registered and logged in	1)Login using username and password 2) Click on each option link on the sidebar	User login and password	Each link should reroute to the appropriate page with the correct user information
13	Add new smart bulb	User logged in, at profile page.	1. Go to profile page 2. Select desired cluster 3. Input Bulb ID 4. Click Submit	Cluster Bulb Name	System JSON object will reveal newly added bulb
14	Add Bulbs to Cluster	Smart bulbs must be previously registered	1. Go to profile page 2. Type in name of desired cluster	SmarBulb ID	System JSON object will reveal the bulb IDs of the associated

			3. Click "Submit"		bulbs
15	Bulb Change Brightness via Slider	Smart bulbs must be previously registered	1. Go to dashboard page 2. Use Individual bulb slider	Bulb choice, Moving Slider to change value	Brightness of individual bulb changes brightness depending on value
16	Bulb Remote Toggle	Smart bulbs must be previously registered	1. Go to dashboard page 2. Press individual bulb toggle button	Bulb choice, Button Press	Individual light turns on or off
17	Cluster Change Brightness via Slider	Clusters must be previously registered	1. Go to dashboard page 2. Choose Cluster from dropdown 3. Use cluster bulb slider	Cluster choice, Moving Slider to change value	Brightness of bulb clusters changes brightness depending on value
18	Cluster Remote Toggle	Clusters must be previously registered	1. Go to dashboard page 2. Choose Cluster from dropdown 3.Press Cluster toggle button	Cluster choice, Button Press	Bulb Cluster turns on or off

12.3 Bugs

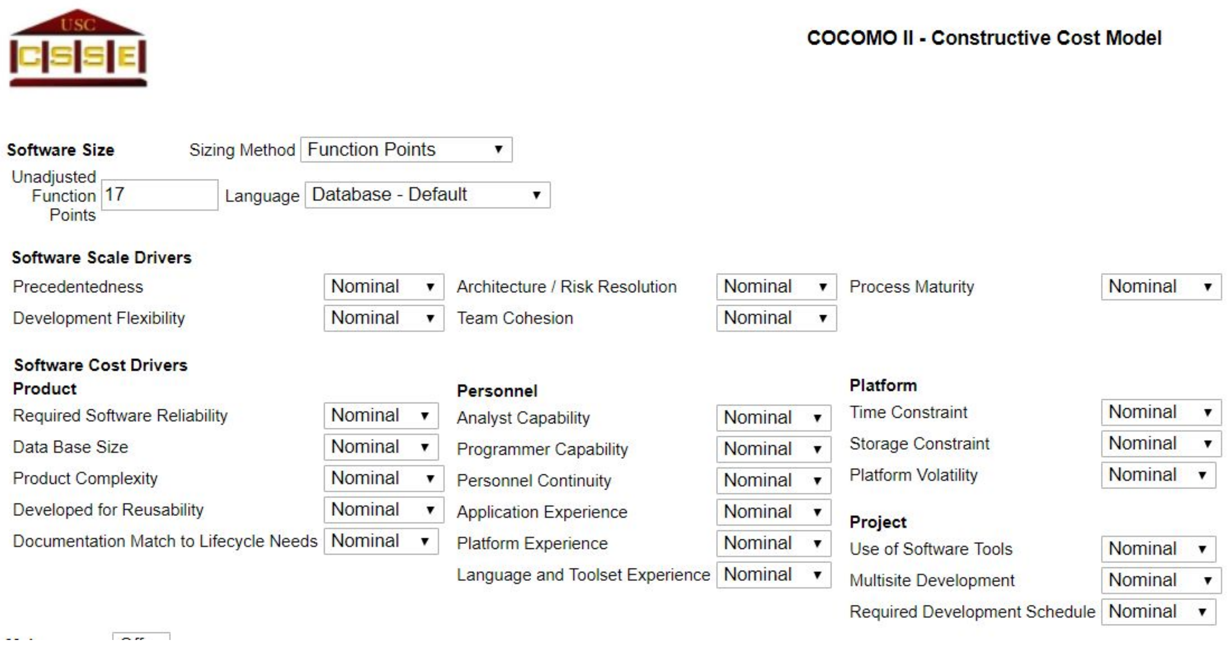
Bug	Test Used	Description of Bug	Action to Fix Bug
Back button not working	10	Back button is unresponsive. Does not route to the previous page	Check link and reference pertaining to the back button in each page
User not automatically logged	11	When the browser is closed and reopened	Keep track of the session using the

		the user remains logged in as opposed to being logged out	username and enforce logout in code when the session is closed
Profile Buttons do not work.	13,14	All buttons on profile nonfunctional.	Integrate “submit” buttons smartbulb API, Integrate “update” buttons with MySQL.
Bulb controls on dashboard nonfunctional	15-18	Button/Sliders for Bulb/Cluster control nonfunctional.	Integrate buttons/sliders with smart bulb API

COCOMO Personnel Cost Drivers

Created using this provided tool: http://csse.usc.edu/tools/COCOMOII.php

13.1 Cynthia Khan



COCOMO II - Constructive Cost Model

Software Size Sizing Method: Function Points

Unadjusted Function Points: 17 Language: Database - Default

Software Scale Drivers

Precedentedness	Nominal	Architecture / Risk Resolution	Nominal	Process Maturity	Nominal
Development Flexibility	Nominal	Team Cohesion	Nominal		

Software Cost Drivers

Product

Required Software Reliability	Nominal
Data Base Size	Nominal
Product Complexity	Nominal
Developed for Reusability	Nominal
Documentation Match to Lifecycle Needs	Nominal

Personnel

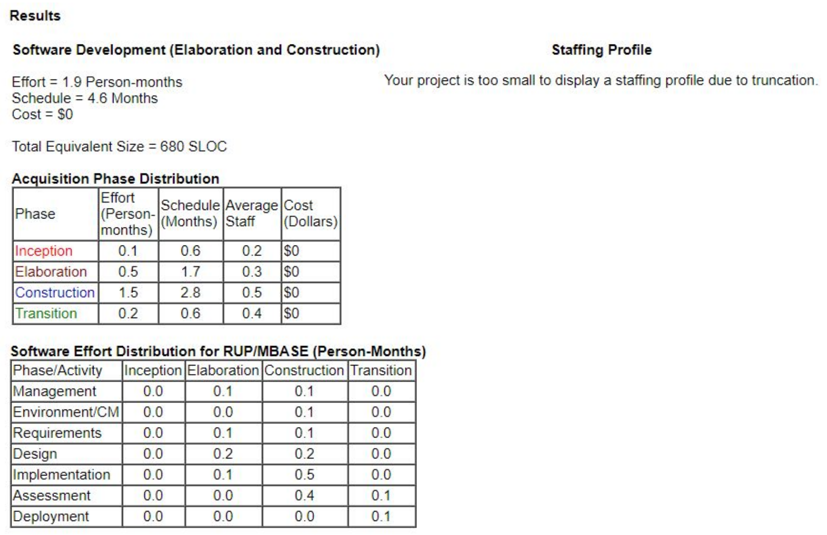
Analyst Capability	Nominal
Programmer Capability	Nominal
Personnel Continuity	Nominal
Application Experience	Nominal
Platform Experience	Nominal
Language and Toolset Experience	Nominal

Platform

Time Constraint	Nominal
Storage Constraint	Nominal
Platform Volatility	Nominal

Project

Use of Software Tools	Nominal
Multisite Development	Nominal
Required Development Schedule	Nominal



Results

Software Development (Elaboration and Construction)

Effort = 1.9 Person-months
Schedule = 4.6 Months
Cost = $0

Total Equivalent Size = 680 SLOC

Staffing Profile

Your project is too small to display a staffing profile due to truncation.

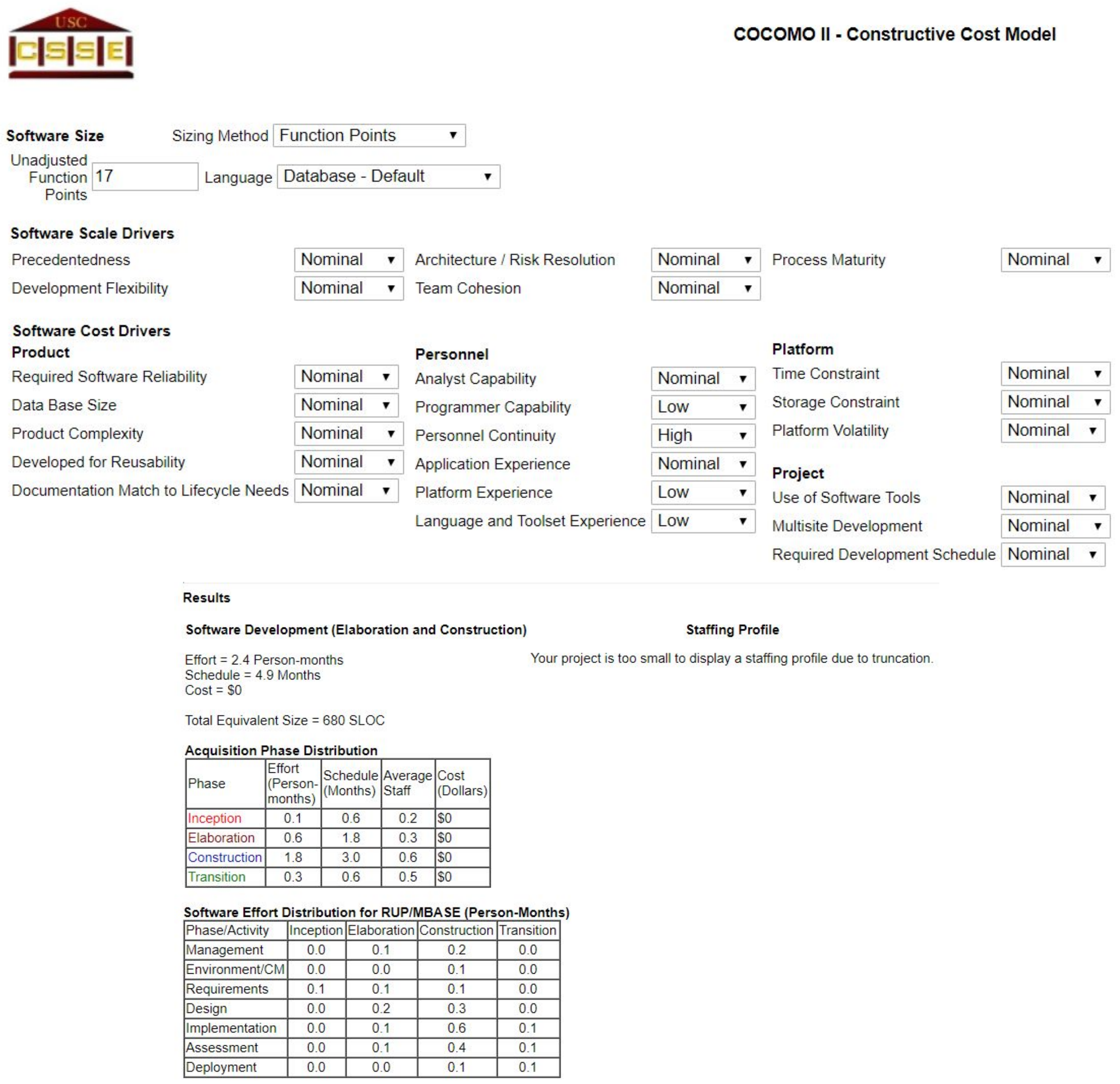
Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	0.1	0.6	0.2	$0
Elaboration	0.5	1.7	0.3	$0
Construction	1.5	2.8	0.5	$0
Transition	0.2	0.6	0.4	$0

Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.0	0.1	0.1	0.0
Environment/CM	0.0	0.0	0.1	0.0
Requirements	0.0	0.1	0.1	0.0
Design	0.0	0.2	0.2	0.0
Implementation	0.0	0.1	0.5	0.0
Assessment	0.0	0.0	0.4	0.1
Deployment	0.0	0.0	0.0	0.1

13.2 Ricky Le



COCOMO II - Constructive Cost Model

Software Size Sizing Method: Function Points

Unadjusted Function Points: 17 Language: Database - Default

Software Scale Drivers

Precedentedness	Nominal	Architecture / Risk Resolution	Nominal	Process Maturity	Nominal
Development Flexibility	Nominal	Team Cohesion	Nominal		

Software Cost Drivers

Product

Required Software Reliability	Nominal
Data Base Size	Nominal
Product Complexity	Nominal
Developed for Reusability	Nominal
Documentation Match to Lifecycle Needs	Nominal

Personnel

Analyst Capability	Nominal
Programmer Capability	Low
Personnel Continuity	High
Application Experience	Nominal
Platform Experience	Low
Language and Toolset Experience	Low

Platform

Time Constraint	Nominal
Storage Constraint	Nominal
Platform Volatility	Nominal

Project

Use of Software Tools	Nominal
Multisite Development	Nominal
Required Development Schedule	Nominal

Results

Software Development (Elaboration and Construction)

Effort = 2.4 Person-months
Schedule = 4.9 Months
Cost = $0

Total Equivalent Size = 680 SLOC

Staffing Profile

Your project is too small to display a staffing profile due to truncation.

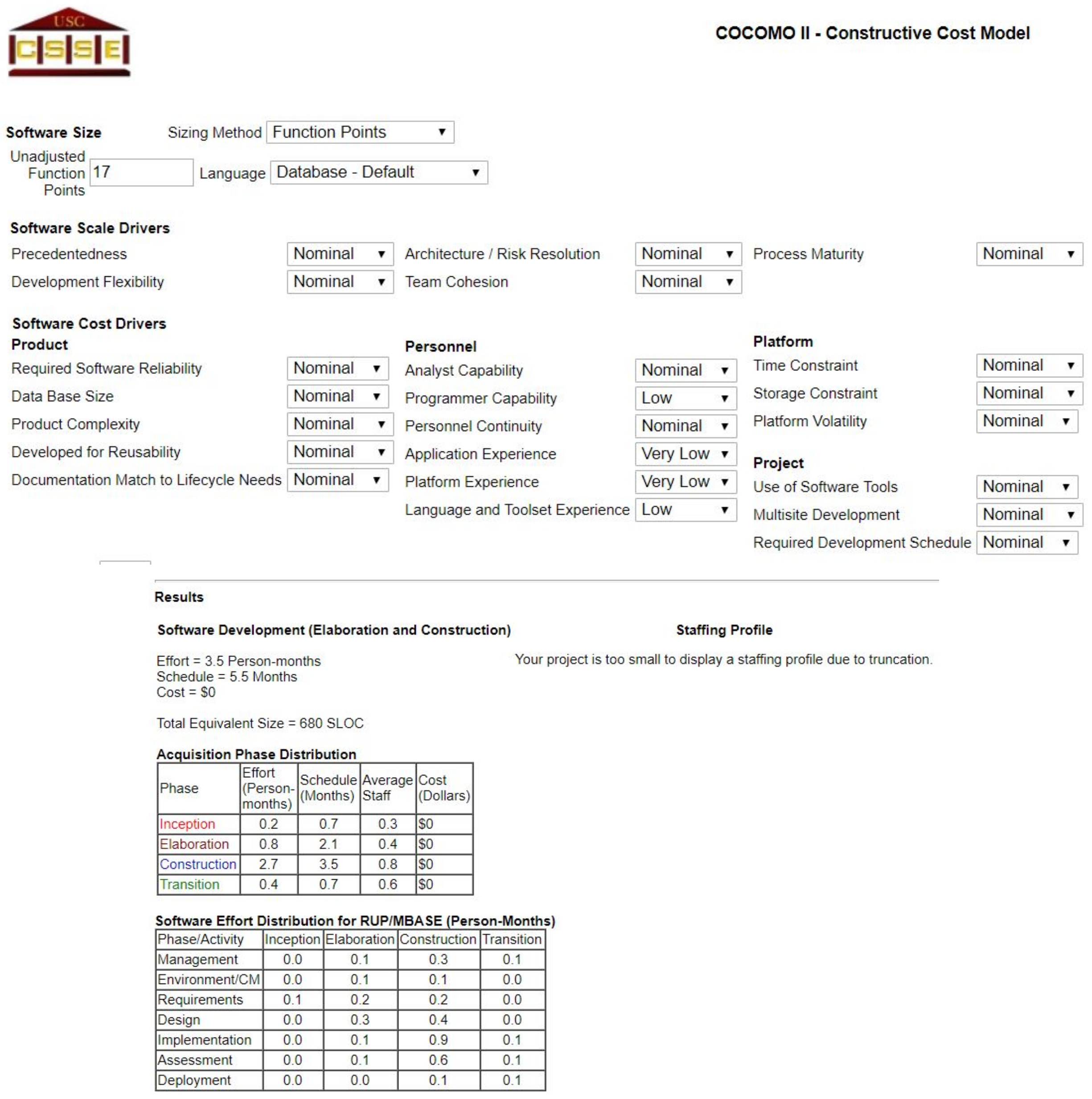
Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	0.1	0.6	0.2	$0
Elaboration	0.6	1.8	0.3	$0
Construction	1.8	3.0	0.6	$0
Transition	0.3	0.6	0.5	$0

Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.0	0.1	0.2	0.0
Environment/CM	0.0	0.0	0.1	0.0
Requirements	0.1	0.1	0.1	0.0
Design	0.0	0.2	0.3	0.0
Implementation	0.0	0.1	0.6	0.1
Assessment	0.0	0.1	0.4	0.1
Deployment	0.0	0.0	0.1	0.1

13.3 Caio Melo



COCOMO II - Constructive Cost Model

Software Size

Sizing Method	Function Points
Unadjusted Function Points	17
Language	Database - Default

Software Scale Drivers

Precedentedness	Nominal	Architecture / Risk Resolution	Nominal	Process Maturity	Nominal
Development Flexibility	Nominal	Team Cohesion	Nominal		

Software Cost Drivers

Product

Required Software Reliability	Nominal
Data Base Size	Nominal
Product Complexity	Nominal
Developed for Reusability	Nominal
Documentation Match to Lifecycle Needs	Nominal

Personnel

Analyst Capability	Nominal
Programmer Capability	Low
Personnel Continuity	Nominal
Application Experience	Very Low
Platform Experience	Very Low
Language and Toolset Experience	Low

Platform

Time Constraint	Nominal
Storage Constraint	Nominal
Platform Volatility	Nominal

Project

Use of Software Tools	Nominal
Multisite Development	Nominal
Required Development Schedule	Nominal

Results

Software Development (Elaboration and Construction)

Effort = 3.5 Person-months
Schedule = 5.5 Months
Cost = $0

Total Equivalent Size = 680 SLOC

Staffing Profile

Your project is too small to display a staffing profile due to truncation.

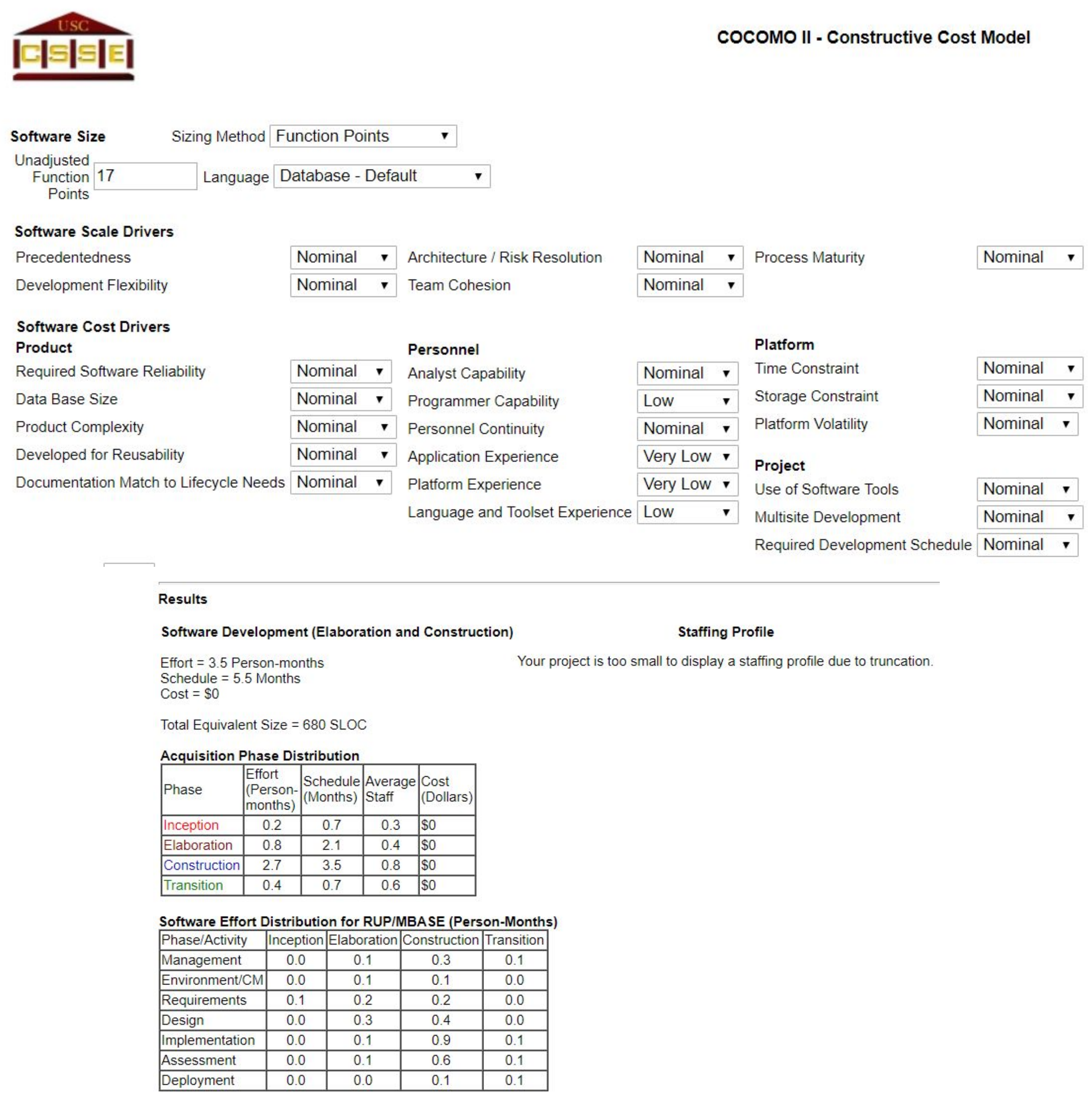
Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	0.2	0.7	0.3	$0
Elaboration	0.8	2.1	0.4	$0
Construction	2.7	3.5	0.8	$0
Transition	0.4	0.7	0.6	$0

Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.0	0.1	0.3	0.1
Environment/CM	0.0	0.1	0.1	0.0
Requirements	0.1	0.2	0.2	0.0
Design	0.0	0.3	0.4	0.0
Implementation	0.0	0.1	0.9	0.1
Assessment	0.0	0.1	0.6	0.1
Deployment	0.0	0.0	0.1	0.1

13.4 Sean Silva



COCOMO II - Constructive Cost Model

Software Size Sizing Method: Function Points

Unadjusted Function Points: 17 Language: Database - Default

Software Scale Drivers

Precedentedness	Nominal	Architecture / Risk Resolution	Nominal	Process Maturity	Nominal
Development Flexibility	Nominal	Team Cohesion	Nominal		

Software Cost Drivers

Product

Required Software Reliability	Nominal
Data Base Size	Nominal
Product Complexity	Nominal
Developed for Reusability	Nominal
Documentation Match to Lifecycle Needs	Nominal

Personnel

Analyst Capability	Nominal
Programmer Capability	Low
Personnel Continuity	Nominal
Application Experience	Very Low
Platform Experience	Very Low
Language and Toolset Experience	Low

Platform

Time Constraint	Nominal
Storage Constraint	Nominal
Platform Volatility	Nominal

Project

Use of Software Tools	Nominal
Multisite Development	Nominal
Required Development Schedule	Nominal

Results

Software Development (Elaboration and Construction)

Effort = 3.5 Person-months
Schedule = 5.5 Months
Cost = $0

Total Equivalent Size = 680 SLOC

Staffing Profile

Your project is too small to display a staffing profile due to truncation.

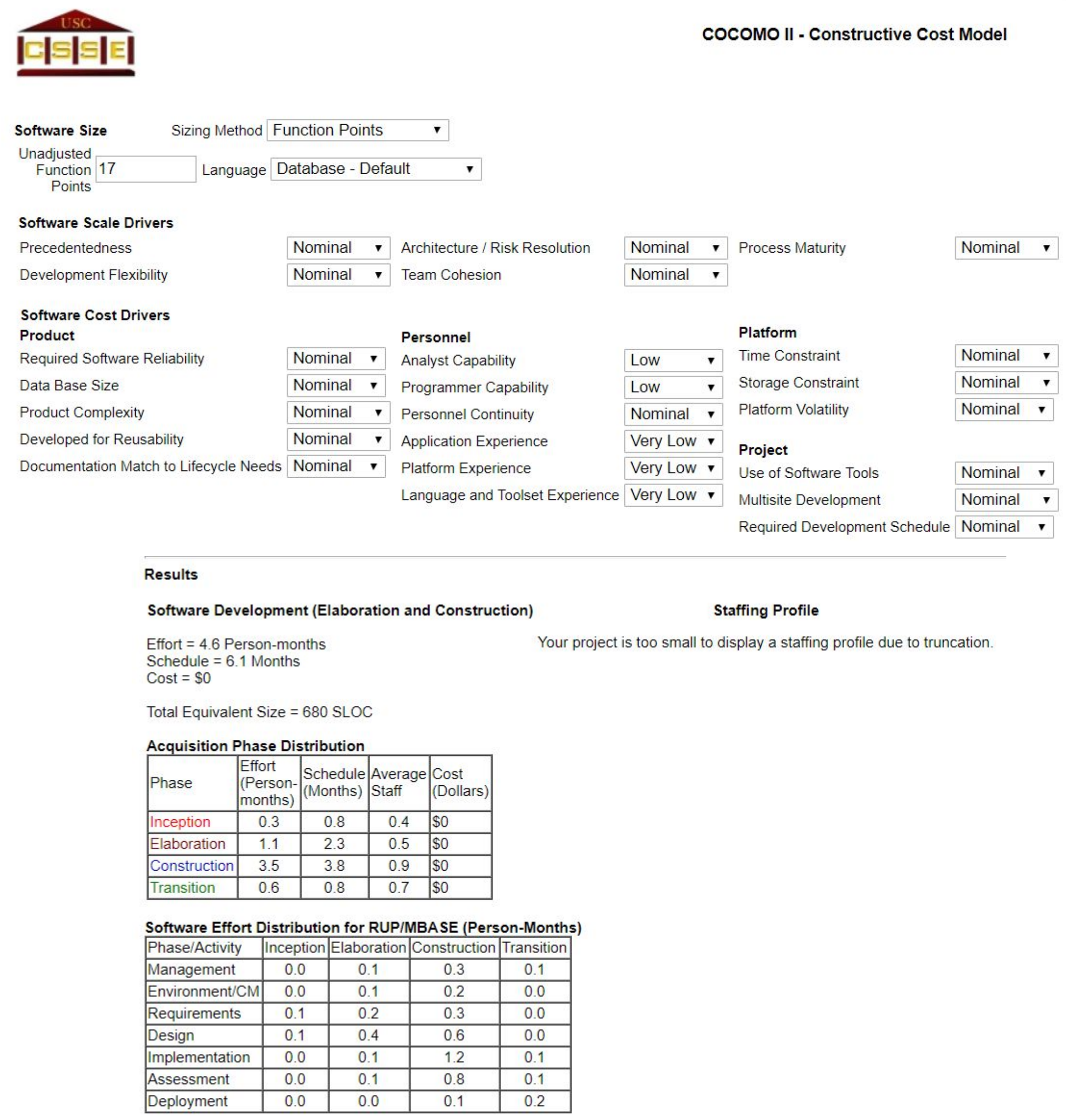
Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	0.2	0.7	0.3	$0
Elaboration	0.8	2.1	0.4	$0
Construction	2.7	3.5	0.8	$0
Transition	0.4	0.7	0.6	$0

Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.0	0.1	0.3	0.1
Environment/CM	0.0	0.1	0.1	0.0
Requirements	0.1	0.2	0.2	0.0
Design	0.0	0.3	0.4	0.0
Implementation	0.0	0.1	0.9	0.1
Assessment	0.0	0.1	0.6	0.1
Deployment	0.0	0.0	0.1	0.1

13.5 Noah Yasarturk



COCOMO II - Constructive Cost Model

Software Size Sizing Method: Function Points

Unadjusted Function Points: 17 Language: Database - Default

Software Scale Drivers

Precedentedness	Nominal	Architecture / Risk Resolution	Nominal	Process Maturity	Nominal
Development Flexibility	Nominal	Team Cohesion	Nominal		

Software Cost Drivers

Product

Required Software Reliability	Nominal
Data Base Size	Nominal
Product Complexity	Nominal
Developed for Reusability	Nominal
Documentation Match to Lifecycle Needs	Nominal

Personnel

Analyst Capability	Low
Programmer Capability	Low
Personnel Continuity	Nominal
Application Experience	Very Low
Platform Experience	Very Low
Language and Toolset Experience	Very Low

Platform

Time Constraint	Nominal
Storage Constraint	Nominal
Platform Volatility	Nominal

Project

Use of Software Tools	Nominal
Multisite Development	Nominal
Required Development Schedule	Nominal

Results

Software Development (Elaboration and Construction)

Effort = 4.6 Person-months
Schedule = 6.1 Months
Cost = $0

Total Equivalent Size = 680 SLOC

Staffing Profile

Your project is too small to display a staffing profile due to truncation.

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	0.3	0.8	0.4	$0
Elaboration	1.1	2.3	0.5	$0
Construction	3.5	3.8	0.9	$0
Transition	0.6	0.8	0.7	$0

Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.0	0.1	0.3	0.1
Environment/CM	0.0	0.1	0.2	0.0
Requirements	0.1	0.2	0.3	0.0
Design	0.1	0.4	0.6	0.0
Implementation	0.0	0.1	1.2	0.1
Assessment	0.0	0.1	0.8	0.1
Deployment	0.0	0.0	0.1	0.2

Self-Reflection

14.1 Slack

Slack was useful in that it allowed us to relay information effortlessly. It also has very good integration with IOS/Android and allowed us to share what was on our phones and devices very smoothly. Slack also has dedicated channels for specific subject matters so the main chat did not get cluttered, and information did not get lost. The ability to mention someone when their input was needed and have them get a notification, was also very much appreciated and used.

14.2 Bootstrap Studio

Bootstrap at first was very helpful in designing the initial look of the website. It quickly become problematic when specific features and functions were tweaked for our specific purpose. In retrospect it probably wasted as much time as it did in saving it.

14.3 PHP

PHP is a must have in any system that wishes to incorporate a database. After a slight learning curve, it quickly become the arms and legs of the website. Realizing how much we would have to use PHP ahead of time and learning it would have been helpful in having the functions done earlier.

14.4 MySQL Manager

MySQL was necessary and very much helpful in: allowing our website to function, holding the information that the machine learning aspect of our system would utilize, holding the information that would be used to populate tableau, and keeping track of the data from the Hue Smart Bulbs. Discrepancies in between using MySql and MySqli within the PHP code resulted in some problems in retrieving data which were eventually cleared up.

14.5 Design Choice

We would still use the same design, but to an extent. If more time was given, we would integrate Tableau Server in order to get a dynamically updating dashboard, which would change the structure of how Tableau interacts with our system, as well as change how the Tableau API would be integrated within our website. We would also add communication between the Philips Hue API with Tableau API to create an interactable graphic that depicts overconsumption of power per area/building. With the current design, there is no interaction between Tableau and the Smart Bulbs, thus we would add structural functionalities within the design. The current design is very modular, so many functionalities can be added. Along with this, due to time constraints and budget,we could not implement a smart thermostat. By implementing a *Nest Smart Thermostat*, the system would heavily improve our existing functions and improve upon our machine learning algorithms/data training set as more information is readily collected.

14.6 Development Model

If given a choice, our project would best be suited with the Agile methodology. And if given a choice between the different flavors of Agile, we would have preferred to adopt Scrum as a method to develop this project. The reasons why a Scrum method best suits our project are:
The different subsystems of our project are more or less independent of each other in the development stage although, in the end, their implementation is only possible when all the subsystems come together. However, since each of the systems can be developed on its own, an Agile method would ensure faster delivery of each subsystem and would also provide the opportunity to revise and make changes to them at the development stage, thus cutting down on cost and time for any changes made. It would also ensure that the end product, a culmination of all the subsystems, would match the client's list of individual functionalities better. Because, if we use waterfall, as we have used for this project, then once the end product is complete, if the client then wants to make changes to one feature that was implemented in one of the earlier processes in the timeline, it might not even be possible since that would mean making changes to all the other subsystems. Also, if we were to discuss the other Agile methodologies for this project, the most common Agile method, XP, would be an overkill for the size of our project. Each member of the team had a different sub system to work on and pair programming would have been futile. All things considered, this project requires a team of 5 to 7 members to meet completion and a sprint cycle would be perfect to ensure management and timely delivery of the end product. Therefore, if given the choice, we think it would be best to develop this project in the Scrum flavor of Agile methodology. What's more, the test-driven methodology of Agile would mean we'd be designing the features specifically to satisfy our tests, giving us something to specifically hone in on and precisely accomplish.

14.7 GitHub

When used to the fullest of its capabilities, GitHub is a powerful tool. In our case, the use of GitHub seemed like an extra unneeded step in our development process. We primarily used Google Drive to keep the majority of our work accessible to all the team members. The drive with the use of slack allowed us to communicate clearly with one another when a part of the project was updated. However, GitHub is great for collaborating projects and files into one place. It is fairly similar to the drive, however it hosts code repositories. It provides developers with a toolset that makes it easy to follow coding practices and allows you to keep all components of a system into one place that all users can easily change, commit then push their updated version. I believe if we started with using GitHub originally rather than using Google Drive, we all would have gained more experience with this important interface, it would have kept some files and key components to our project in a more organized manner, and lastly provided us with a safe peace of mind knowing our project was in an environment that could support our system files.

14.8 Vision Achievement

Our project met perhaps 75% of its original goal. We were able to create a stable server and website with data visualization via Tableau charts as well as wire the user registration and login features through our MySQL tables. Our project also met the nonfunctional requirements we set up in advance of being aesthetically pleasing and reliable. Our alert system, while existing only via a user actually connected to the site rather than implemented via email, was in functioning order. We had typical login functionalities. Where we failed was at one of our core functionalities the bulb controls implementation. This feature was fundamental to our project as it separated us from existing market competition. What's more, the platform did not actually include any real back-end code to do with bulb and cluster registration as we were so stuck on attempting to connect the controls to the bridge. Had we more time (a day or two), there's no doubt these would've been easily implemented, but a deadline is a deadline.

14.9 Planning and Scheduling Analysis

Initially, due to the unfamiliarity our project manager had with managing others' work and with understanding the degree of capability and effort his group members had, planning was quite difficult. What's more, the learning curve on some technologies our system relied upon (nodeJS, node package manager, JavaScript, Tableau, Philips Hue API, browserify) turned out to be quite substantial. Noah could've done a better job of emphasizing the importance of set deadlines for assigned tasks for which other tasks were dependent upon. Once SmartSheet usage was added to the project, things ran MUCH smoother. Reposting the same text in Slack turned out to be highly inefficient.

In terms of time estimation, we greatly underestimated the learning curve time. We should've allowed more time for the actual bulb registration and coding. We ran into issues insofar as understanding how the GSU network assigns IP addresses, how the subnets of Aderhold (where our demo took place) are organized, and using the node module huejay in our browser. We even ran into a bit of a security problem with the GSU network due to the fact that the bridge discovery feature wouldn't initially work on the GSU network, causing us to have to search for another third-party solution, the Advanced IP Scanner. Huejay had already codified methods for SmartBulb registration and functionalities, but to use it we had to use either browserify or webpack to arrange our JavaScript files and node modules in a specific order so as to allow the node server-side modules to be used within the client-side browser. In retrospect, simply analyzing the Philips Hue developer site's provided API GUI via web developer tools in Chrome would've and making our own methods without the use of someone else's GitHub would've actually saved us time and may have allowed us to actually implement the intended features. Unfortunately, when attempting to meet deadline, we developed a sort of tunnel vision and should've tried to explore additional implementation options.

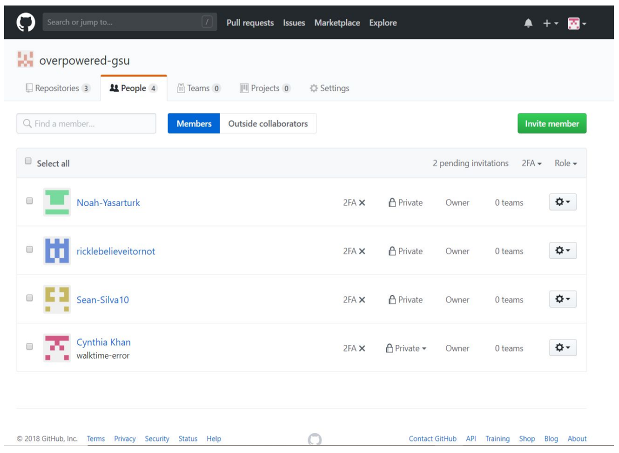
14.10 Consideration of a Prototype Approach

Because our user requirements came from the developers themselves, WalkTime Error did not truly need a prototype to help explain our abstract vision. We all seemed to understand the core idea of what the platform would do, how the site would be structured, and what implemented features would look like. The core of our issues in term of our discussion over what certain functionalities would look like when made concrete had to do with how data collected from the SmartBulbs would be inputted into Tableau and our prediction model- the rate, where the information could perceivably come from, etc. A prototype would not be useful for this.

Appendix

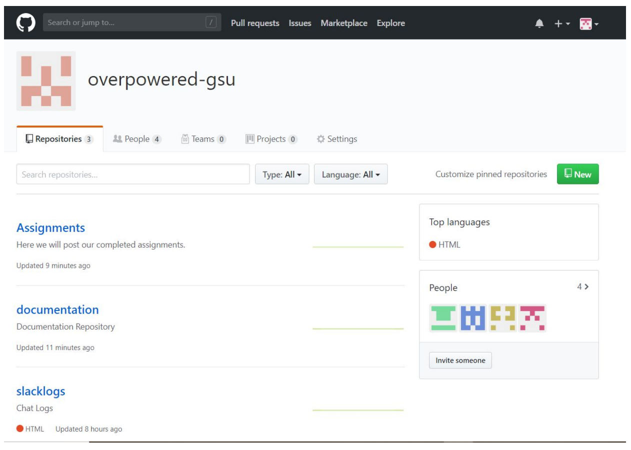
15.1 Appendix 1.0: GitHub

GitHub link: https://github.com/overpowered-gsu



Noah-Yasarturk	2FA ✗	Private	Owner	0 teams
ricklebelieveitornot	2FA ✗	Private	Owner	0 teams
Sean-Silva10	2FA ✗	Private	Owner	0 teams
Cynthia Khan walktime-error	2FA ✗	Private	Owner	0 teams

© 2018 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub API Training Shop Blog About



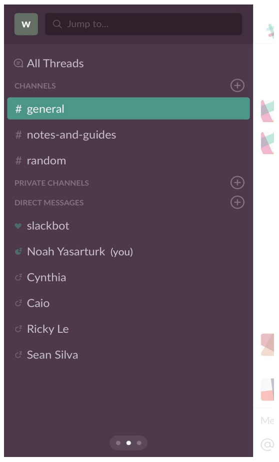
Assignments
Here we will post our completed assignments.
Updated 9 minutes ago

documentation
Documentation Repository
Updated 11 minutes ago

slacklogs
Chat Logs
HTML Updated 8 hours ago

15.2 Appendix 1.1: Slack Logs

Slack Logs link: https://github.com/overpowered-gsu/slacklogs


Jump to...

All Threads

CHANNELS

general

notes-and-guides

random

PRIVATE CHANNELS

DIRECT MESSAGES

slackbot

Noah Yasarturk (you)

Cynthia

Caio

Ricky Le

Sean Silva

15.3 Appendix 1.2: Use Case Diagram

Use Case Diagram Link: http://bit.ly/2KhH3J3

15.4 Appendix 1.3: Class Diagram(s)

Class Diagram(s) Link: http://bit.ly/2MwHIDm

15.5 Appendix 1.4: Sequence Diagrams

Sequence Diagrams Link: https://bit.ly/2tFKM9d

15.6 Appendix 1.5: Entity-Relationship Diagram

ERD File Link: https://bit.ly/2K95HaN

15.7 Appendix 1.6: Website Files

Website Files: http://bit.ly/2lK9wc4