Overpowered

Summer 2018

Walktime Error

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

6/8/18

Planning and Scheduling

Milestones	Principal Tasks	Software/Hardware Requirements
Realistic utility bill data simulation	1.) Obtain real power bills/research typical power consumption to define lower/upper limits for random generation 2.) Randomly generate Strings and Floats 3.) Format as csv 4.) Feed csv to server	Corporate Utility Bills Public datasets: https://www.data.gov/energy/
Create/configure server to be database (backend development) Database Server	1.) Secure a GSU server 2.) Create permissions for us to access as admins 1) Create ER diagram 2) Create Database schema 3) Test Table relationships/ dependencies 4) Insert data 5) Test database connection/ functionality to support the rest of the software	GSU Server: ● OS = Windows ● CPU = 4 core ● RAM = ● Storage = 1 GB Microsoft SQL Server
Design website aesthetically	1.) Research competition 2.) Draw a prototype digitally/photoshop of the individual pages 3.) Create the HTML skeleton 4.) Apply CSS styling as per customer directions 5.) Implement JavaScript or PHP functionalities	HTML, CSS, JS, PHP, Sublime Text, Eclipse/IntelliJ

	to connect to Database or other APIs	
Machine learning algorithm development	1.) Determine what information the platform will be tracking and the client is concerned with 2.) Build Prediction models based on the training dataset from the database 3.) Research Python packages already available (Mussa is an expert on this, will help) 4.) Use the prediction model to predict future outcome given test conditions 5.) Send results to website interface for client to view	Python/R
Configure/integrate smartbulb API to feed database power consumption information	1.) Buy smart bulb 2.) Determine format of wattage info output; re-format to feed server, maintaining individual bulb ID 3.) Determine the ability to create clusters/groups with individual bulbs to classify as rooms/hallways 4.) Integrate with database	Phillips Hue kit (smart bulb and bridge) Javascript
Configure/integrate smart bulb API to allow remote power operation (on/off switch)	1.) Buy smart bulb 2.) Determine on/off function from API 3.) Determine ability to change intensity of	Phillips Hue kit Java/Javascript

	bulbs (changing wattage output) 4.) Develop solution to link smart bulb(s) to back-end 5.) Link this to front-end web platform	
Integrate data visualization tools to web platform	1.) Create graph/ charts depicting overall building power consumption throughout day, week, month, etc. 2.) Create real-time graphic depicting power consumption by cluster 3.) Create bar chart integration depicting comparisons between clusters 4.) Create line graph integration depicting power usage over several time scales 5.) Take user input and display results	Tableau, HTML
Develop client alert system	1.) Define upper parameter for power overconsumption by either output of ML or client specification 2.) Send email (containing link to web platform) to client should parameter be breached	C++ or PHP, Database mail from MS SQL Server
Develop client registry parameters and definition	1.) Log in, register new client/user on web platform 2.) User authentication system	Database, Web API

	3.) Require client to register all smart devices / Define clusters	
Integrate front-end to back-end	1) Integrate Database, API Features, and all functionalities of website to dashboard	PHP, Javascript

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	Maintain constant communication so as to ensure we're all aware of each others' responsibilities should they need to be redistributed
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

Project leader- our project leader has a tendency to assume too much responsibility and may be slow to require team members to complete their tasks on schedule	Due to the democracy we established, another group member can step up and assume some more responsibilities as well.
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

Task	**Effort** (person-days)	**Duration** (hours)	**Dependencies**	**Person**
Database Setup	**1.86**	6	Server setup Data	Cynthia Khan
Learn Tableau	**1.3**	5	Database and Website setup	Noah Yasarturk, Caio Melo, Sean Silva
Website Template	1.3	5	Server setup	Caio Melo
Smart Bulb API Integration	4	5-10	Smart Bulb Delivery Time, API Learning Curve	Ricky Le, Noah Yasarturk
Website Functions Integration	2	4	Functional database, Smartbulb API, Dashboard display	Noah Yasarturk, Sean Silva, Ricky Le, Cynthia Khan
Dashboard Creation	6	8-12	Database being created; Machine learning being able to cypher the data; learning Tableau	Sean Silva, Noah Yasarturk
User Registry	1	5	Website (front-end/back-	Cynthia Khan, Ricky Le, Caio

			end) being functional	Melo
Prediction Modelling Integration	2.86	7	Acquire test data and training dataset	Cynthia Khan, Ricky Le, Noah Yasaturk
Realistic Data Simulation	2.9	8	Power Company Client Research	Noah Yasaturk, Sean Silva

Scheduling

Date	Noah	Ricky	Cynthia	Caio	Sean
6/19	Research public datasets for energy consumption	Learn smart bulb API functions Discover formatting for wattage output data	Set up Server Research on Microsoft SQL server functionalit ies Create Database Insert public dataset into relevant tables	Design Website Template	Learn Tableau Functionalities; create theoretical client building structure
6/20					
6/21					
6/22					
6/23					
6/24					

6/25	Develop algorithm to reliable feed realistic utility bill info	Integrate smart bulb with server	Determine ML Packages		Integrate smart bulb with server
6/26				Develop website front-end framework/design	
6/27	Link data simulatorto server	Scale up smart bulb into theoretical building network; define clusters	Develop prediction model		Develop Real-Time Smart Bulb cluster GUI framework
6/28					
6/29					
6/30					
7/1					
7/2	Develop website back-end framework (platform structure, template for functionalities)	Integrate Smart bulb GUI with server			Integrate Smart bulb GUI with server
7/3					

7/4			Develop website back-end framework (email alert system, link to database)		
7/5		Develop Client Registry System		Integrate website front and backend, connect to server/database	Integrate Tableau visualization tools to website
7/6					
7/7					
7/8					
7/9	Integrate website front and backend, connect to server/database	Develop user-query system	Develop budget framework- include cost/watt		
7/10					Integrate website front and backend, connect to server/database
7/11					
7/12					
7/13	TESTING	TESTING	TESTING	TESTING	TESTING
7/14	(Margin for error)				

7/15					
7/16					
7/17	PRESENT	PRESENT	PRESENT	PRESENT	PRESENT

Monitoring and Reporting Mechanisms

We'll make use of Jira 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?

Collaboration

GitHub link: https://github.com/overpowered-gsu
Slack Logs link: https://github.com/overpowered-gsu/slacklogs
YouTube Video link: https://youtu.be/jdOaPTeZmjA

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

Refer to Appendix Section 1.0 for Context Diagram

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

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

Requirement #: 3
Use Case: Register New User, User Login, Database Functionality, Smartbulb API Functionality
Date: June 15th, 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 and Password, which then leads to submitting Utility data and Building Layout Schematics, and registering of smart devices.
Outputs: Profile information is saved into Database, which allows future login to access website dashboard.
Related Requirements: 2, 4, 5, 7

Requirement #: 4
Use Case: Register New User, User Login, Dashboard Display Functionality, Database Functionality, Smartbulb API Functionality, Machine Learning Functionality
Date: June 15th, 2018
Introduction: Website Design Functionality
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: Database Information (Profile information + Utility information), Smart Device API functions
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

Requirement #: 5
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

Requirement #: 6
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

Requirement #: 7
Use Case: Register New User, Dashboard Display Functionality, Smart Bulb API Functionality
Date: June 15th, 2018
Introduction: Configure/Integrate Smart Bulb API Features
Rationale: Using the Phillips 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:
Outputs: Remote Control of Lighting, Grouping/Clustering of Individual Lights, Timed toggles, Integration within Website Features
Related Requirements: 3, 4, 5, 6

Requirement #: 8
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

Refer to Appendix Section 1.1 for Use Case Diagram.

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

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

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

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

Use Case Name: Smart Bulb API Functionality
Summary: Using the Phillips 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.

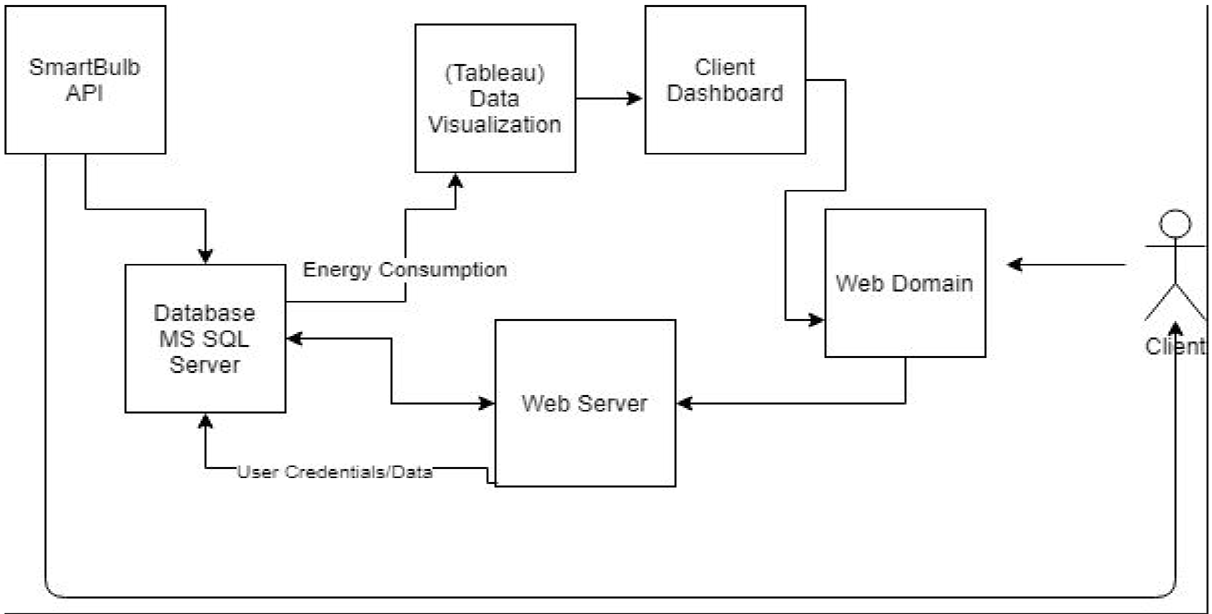
Exception Paths: Bulb information regarding wattage is unreadable to the machine learning system, which function must be re-evaluated to output desirable information. The room/cluster of bulbs have been remotely toggled while there is occupancy, which occupants can manually reset by toggling the wall 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 Phillips Hue API is open and readily available to use.
Precondition: Access to the Phillips 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

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

Appendix

Appendix 1.0: https://bit.ly/2yhQouQ



Appendix 1.1: https://bit.ly/2MtemGN