



College of Engineering

CS CAPSTONE REQUIREMENTS DOCUMENT

OCTOBER 27, 2017

A SCALABLE WEB APPLICATION FRAMEWORK FOR MONITORING ENERGY USAGE ON CAMPUS

PREPARED FOR

OREGON STATE OFFICE OF SUSTAINABILITY

JACK WOODS

Signature

Date

PREPARED BY

GROUP 57

THE DREAM TEAM

DANIEL SCHROEDER

Signature

Date

AUBREY THENELL

Signature

Date

PARKER BRUNI

Signature

Date

Abstract

This document outlines the requirements set forth by The Office of Sustainability as to what should be included in the final product of our project.

CONTENTS

1	Introduction	2
1.1	Purpose	2
1.2	Scope	2
1.3	Definitions, acronyms, and abbreviations	2
1.4	References	2
1.5	Overview	2
2	Overall description	3
2.1	Product perspective	3
2.2	Product functions	3
2.3	User characteristics	3
2.4	Constraints	4
2.5	Assumptions and Dependencies	4
2.6	Apportioning of Requirements	4
3	Specific Requirements	4
3.1	External Interfaces	4
3.2	Functions	4
3.3	Performance Requirements	5
3.4	Logical Database Requirements	5
3.5	Design Constraints	6
3.5.1	Standards Compliance	6
3.6	Software System Attributes	6
3.6.1	Reliability	6
3.6.2	Availability	6
3.6.3	Security	6
3.6.4	Maintainability	6
3.6.5	Portability	6

1 INTRODUCTION

1.1 Purpose

The purpose of this document is to outline the project and all associated information. Furthermore, it will outline how the end product will be used, how it was developed, and all resources and documentation referenced and used during development. This document will also describe the applications target audience, user interface, and any hardware/software requirements. Finally, it will define how our client, team, and users will see the final product and related functionality.

1.2 Scope

Primarily, the scope will pertain to the Scalable Web Application Framework for Monitoring Energy Usage on Campus project. The project will display different collections that reports energy usage for campus buildings across campus in a user-friendly manner. Each of these collections will be customizable, with the ability to add or remove relevant energy readings to user desire. There will be three main permission levels: Guest, User, and Admin. Guest access will allow anyone with a link to view a specific dashboard with no ability to edit. User will be able to create their own collections from pre-existing data. Admin will be able to add, remove, manage, and change anything on the dashboard application. This will allow the Sustainability Office, Oregon State University affiliates, and the general public to each have permissions to view energy usage.

1.3 Definitions, acronyms, and abbreviations

Term	Definition
OSU	Oregon State University
MEAN stack	MongoDB, Express, AngularJS, Node.js
JSON	JavaScript Object Notation
AcquiSuite	Data Acquisition Servers responsible for metering energy usage.
Collections	User Interface dashboard that displays assortment of blocks
Blocks	Charts/graphs of building data

1.4 References

The references are: Blank for now

1.5 Overview

The remainder sections of this document provide an overall description, including product functionality, assumptions, and specific requirements. Section 2 gives functional requirements, data requirements, and assumptions made while designing Scalable Web Application Framework for Monitoring Energy Usage on Campus. Section 3 gives the specific requirements of the end product as well as discuss the external interface requirements along with functional requirements.

2 OVERALL DESCRIPTION

2.1 Product perspective

The application that we aim to create will essentially take the place of the current application that is used by the Oregon State Sustainability office. The current application in place has been developed by a company called Lucid that was contracted by the Sustainability office to gather and display information about Oregon State buildings energy consumption to make better infrastructure decisions related to energy consumption. The application makes use of energy monitoring meters dubbed AcquiSuite meters (implemented by the company Obvius) that are already in place inside the buildings, with some buildings containing multiple meters monitoring various distinguishable sections of the building. The current application in place processes the data from the meters and organizes it in an easily interpretable and intuitive way. A user may view this data presentation via web browsing devices such as computer systems or mobile smartphones. Our goal for this product is to mimic the functionality of the current system in place with more efficient alterations to the design. The new product will aim to remove unnecessary features from the design and add new functionality and features that are more desirable for the application users. The product will contain a database that will gather live data from the energy meter data interface as well as be updated with past data about OSU buildings energy consumption. The energy data will be presented two separate UIs. One UI will focus on a clean general public interface that has relatively simple information navigation features, the other UI will be a more customizable dashboard of information presentations such as graphs or charts that will be accessible only to approved users.

2.2 Product functions

User accounts will be able to be created via automatically generated invitation emails via batch emails or individual emails. User accounts can be given either administrative Permissions, or user permissions. Users can create personalized “collections” to view their own data. Administrators can create/edit public “collection”, as well as perform general administration tasks. All users can opt in/out of email and text message alerts for offline buildings or exceedingly high energy usage alerts. All users can opt in/out of monthly or weekly energy usage reports (delivered via email).

The application will allow users to create customizable dashboards dubbed Collections that will contain easily adjustable blocks of building data presentations content (graphs, charts, etc.) that will contain live data as well as past data. These blocks of data will be the basic building blocks for the dashboard and will provide data from any building or building subspaces from any time period (including live data updates). They will also provide various graph types, building energy efficiency rankings, and data trends.

Each OSU building that contains the energy monitoring meter(s) will have a specific, non-customizable page that will display general information.

Administrators of the application will have the ability to add, remove, or edit entire buildings profiles, building subspaces, or individual meters.

2.3 User characteristics

There are two general user interfaces that will differ in user experience. A user that will be using the general public UI will not need to know any specific information about the application to navigate the various energy data presentations. A public user will be able to intuitively navigate the UI to their discretion.

An administrator class user will need to know the basic tools of the application to create the dashboard that is specific

to their profile and needs. An administrator will likely need to use the information for more critical objectives and will be allowed the freedom to control parts of the website as well as have access to more specific energy data within the application. An administrator will likely not need extensive training to use this application for more specific purposes as the administrator UI will be designed to be intuitive to navigate.

2.4 Constraints

The application will be entirely dependant on the energy metering system. If the energy metering system is drastically changed or removed, the application will not be functional. Live data updates will be limited to a granularity of 15 minute intervals. The energy monitoring meters are capable of providing a granularity of up to 15 second intervals but it is not necessary for the purposes of the application. In the future, the application may be altered to achieve a smaller interval of granularity if desired. Meter data within the application will also be limited by the functionality of the meters themselves. Should a meter malfunction, the energy data will not be gathered which may cause some of the data presentations to deviate from expected data. The deviation amount will depend on the number of malfunctioning meters and the timeframe that the meter(s) was malfunctioning. The use of the application will require a compatible web browsing interface, which may be limited to browsers such as Firefox, Chrome, and Internet Explorer. The functionality of the application will depend on those applications performing as expected. The application may fail if those browsers are altered by their associated developers in a way that the application does not account for and/or if those browsers do not have proper internet access.

2.5 Assumptions and Dependencies

2.6 Apportioning of Requirements

Future versions of the application may include features such as cost tables, automated electronic invoice generation, energy billing analysis capabilities, budget analysis capabilities, and mobile energy data entry.

3 SPECIFIC REQUIREMENTS

3.1 External Interfaces

AcquiSuite data acquisition servers [meters] made by Obvious AcquiSuite meter Used for collecting electric, water, gas, steam, and other energy parameters over the web Meters connect through IP-based applications Data can be reached anywhere with an internet connection as long as the AcquiSuite is online Data will be collected every 15 minutes Our application will attempt to receive this information and convert it to JSON for insertion into the MONGODB database

3.2 Functions

- The system shall collect data from AcquiSuite data acquisition servers, validate the data, and store it into the database. The data will need to be interpreted and converted to JSON to be properly stored into the corresponding database table. There should be input validation measures in place to monitor incoming data and protect from malicious injections.
- The system shall retrieve data from the database to create graphical representations on user dashboards. These graphs and charts should update in real time as new data is retrieved from the AcquiSuite data acquisition

servers. The database will return a JSON object full of data that will be parsed and fed into a Javascript graphics framework to create visualizations.

- The system shall have a permissions system with 2 to 3 different user roles. A user can have administrative roles and be able to add Acquisuite meters, buildings, and other objects into the database. A user can be a general user that can view the public dashboard and browse public stories. A user can be a special user, for example the head of a campus department, that can create their own stories that pertain to their relative departments and buildings.
- The system shall generate emails that allow users to sign-up. This email will have a way of granting a user role during sign up so the users is granted specific permissions based on the sign-up link they received.
- The system shall have a form to have users input data when they create a login profile. The user will be stored in the database with a primary key, name, username, password, permission role, and other user attributes. This sign-up form will have data validation on all input parameters to protect against malicious injections. This form should also validate that the user has not already signed up and is not already a part of the system.
- The system shall allow users to customize dashboard layouts in a grid-based orientation. A story page should have a customizable layout where a user will be able to add different blocks and graphs pertaining to information that is relevant to them.
- The system shall have a general page for each building in the database with a uniform dashboard layout. This page will have the same design for each building and should display a series of graphs and charts that outline energy usage for each building.
- The system shall allow users to create stories which are collections of dashboards. These stories are meant to bring related buildings and datasets together into intuitive groups, for instance Residence Halls or Engineering Buildings.
- The system shall allow blocks to combine data from multiple buildings or database entries and apply filters. The blocks will be able to be manipulated based on time constraints.
- The system shall have the ability to rank building based on energy used.
- The system shall be able to download building data as a CSV file.

3.3 Performance Requirements

The web application should be able to scale up to as many buildings are on campus. As new buildings are constructed, or current buildings are metered with AcquiSuite data acquisition servers, an administrative user should be able to add it to the database and use it stories and dashboards. [there should be a metric about number of users. Based on server size in Sustainability Office?]

3.4 Logical Database Requirements

Schema to be added soon

User - {Primary Key, First Name, Last Name, Username, Password, Permissions Role...}

Building - {ID, Name, Type, Energy, Water, Etc}

Collection - {Name, Foreign Building Key 1, Foreign Building Key 2.. Etc}

