Energy Effeciency Center Website

Software Design Description

Garrett Amidon, SR Kanna, James O'Neal
Team #36, Team WinniethePooh

May 4, 2017

Abstract

OSU's Energy Efficiency Center help manufacturing and industrial companies increase their productivity and reduce their energy footprint by producing reports for energy and productivity recommendations. These reports, projects and funds are maintained in their website. The website has been developed and maintained by several programmers. As a result, the website has become disorganized, difficult to update and use. In order to remedy these issues we will design a secure, user friendly website with good code practices for the Energy Efficiency Center. Furthermore, the website is not accessible from mobile devices which decreases productivity while on the job site. With enough time, we would like to create a secure mobile app for the client which they are able to remotely access. This document will show the options available to fix these issues and which we think are best.

Contents

1 Overview

1.1 Scope

This document will discuss the development of the design elements. It will provide a detailed description of nine elements. These elements make up the core features of the system. It will include the connections between functionalities, specific technology, and viewpoints for each feature.

1.2 Purpose

This document specifies content and organization of the software design attributes. It will include explanations, implementation tools, and multiple viewpoints for each design element. It will outline the important aspects of the solution.

1.3 Intended Audience

This document is intended for the developers, clients, and managers of the Energy Efficiency Center project. The developers will use this document to guide them through the selection, organization and presentation of the design information. Clients and managers can use the design document as a guide for the future development process.

1.4 Conformance

The design documentation conforms to clauses 4 and 5. Requirements are signified by shall.

2 Definitions

For the purpose of this standard, the following terms and definitions apply.

Agile: a software engineering project method style
Waterfall: a software engineering project method style
User Stories: A tool in Agile to understand what prioritize the user's needs.
Energy Efficiency Center: The clients
Security: defense of computers against intrusion and unauthorized use of resources
Login: an act of logging in to a computer, database, or system
User Interface: the means by which the user and a computer system interact
Projects and Tasks: projects and tasks must be added, edited, and viewed
Viewing User's Work Hours: employee work hours must be recorded and restricted
Time Clock: employees record number of hours while specifically assigning their hours to a task
Employee Records: Display and edit employee's information

3 Conceptual Model for Software Design Descriptions

3.1 Software Design In Context

This section creates conceptual models for the intended audience. The conceptual model will act as a guide for the developers and can be referenced by clients and managers for a high-level understanding.

Fig. 1: above is a diagram depicting the decencies and connections between design elements.

3.2 Software Design Descriptions Within the Life Cycle

Login credentials are sent to the database to be verified. Upon valid response the login page will redirect to the homepage. The profile homepage will provide navigation to all other pages. The projects and tasks page will query the database for relevant information. The projects and tasks page will include access to clock functionality. Employee information also needs to connect with projects and tasks in order to display relevant data. All three functions interact with the database to read and update data.

4 Design Description Information Content

4.1 Introduction

The required design elements are:

1) Security
2) Login
3) User Interfaces
4) Projects and Tasks
5) Viewing User's Work Hours
6) Code Readability
7) Database Management
8) Time Clock
9) Employee Records

They are described in detail in the remainder of the clause.

4.2 Security

Security should be designed alongside the software. Threats to consider and our criteria for evaluation include malicious SQL injections, XSS, error messages, server side validation, passwords, file uploads are some of the main threats to consider. SQL injections can steal sensitive information or delete sensitive information by inserting sql commands into user inputs. Cross-site scripting attacks occur when malicious scripts are inserted into prompts. Therefore user input must be completely sanitized in javascript, php and html by escaping script tags, special characters, and checking for industry standards such as UTS-8 etc. We should also perform several testing rounds to ensure that our database and website are secure. There are a few free softwares available which check for common website vulnerabilities. One such client approved software is beyondsecurity..

4.3 Login

Login in websites should prevent against common attacks such as sql injections, cross site scripting, brute force attacks. The best method to prevent against these attacks is a combination of secure login can be obtained via php and mysql. This kind of method prevents against sql injections, cross site scripting, brute force attacks etc. I am familiar with the industry standards of php and sql, so this will the most optimal combination to use.

4.4 User Interfaces

Users expect aesthetics and ease of use when interacting with web applications. CSS tools like Bootstrap, Foundation and Boilerplate provide structure, navigation and aesthetic appeal on all web platforms. All three are responsive web tools and all three provide HTML and CSS frameworks for forms, buttons, Javascript extensions, typography. Bootstrap and Foundation's offerings are more extensive than Boilerplate's. All three are supported by multiple browsers, but Bootstrap and Foundation are the most extensive. However, Boilerplate is the most lightweight platform option. Bootstrap has a large open-source platform and is geared towards engineers to use and contribute to. Foundation is primarily used by large corporations. I am also extremely familiar with Bootstrap, so there would be a minimal learning curve or time wasted. Under the categories of structure, aesthetics, time and responsiveness, I recommend using Bootstrap.

4.5 Projects and Tasks

A gantt chart for visualizing projects and tasks can be displayed using DHTMLx. This Javascript library is available to use for free with the GNU GPL license. It includes a connector which dynamically updates a database when changes are made. This allows user interactions to update the current object in real time. An additional advantage of DHTMLx is its documentation. The application has a large source of tutorials and documentation on all of its features. This will allow future programmers to implement additional functionality with ease. The DHTMLx library is stored on the web server in order to be used by the necessary files. It is also cross browser and platform enabled making it the most portable choice.

Projects and tasks are vital to the operation and organization of the EEC. In order to stay on track and be productive employees need to keep projects and tasks organized. Additionally, they need a clear view of overall project timelines and the status of specific tasks. All this information needs to be easily viewable and editable. To accomplish this we will implement a DHTMLx gantt chart which can interface with our database. This chart will provide task hierarchy and a visual representation of projects. Additional information about each project and task will be viewed outside of the chart.

4.6 Viewing User's Work Hours

An HTML table is a highly versatile way of viewing users work hours. The biggest advantage to using HTML is that it can be customized using CSS. This means the table will have the same appearance as the rest of the website. An HTML table can be populated directly from a database and the layout of the fields is highly customizable. This is also a good choice for mobile devices as it will display properly on mobile browsers.

Displaying employee hours is important for both the employees and their supervisor. Clear visual representation of hours worked on a task allows verification of correct input. Additionally they will not want a complex or time consuming process for managing and updating their hours. To accomplish this, The table will show what project an employee has billed hours on and how many total hours they can bill. This will allow the employee to easily recognize when they have exceeded the hour limit for a specific task.

4.7 Code Readability

An option for improving software quality is Codacy. This software is an automated code review platform. It has multiple metrics which can be turned on or off depending on the type of analyzation that needs to be done. Some of the possible metrics are accuracy, complexity, and coding style. It is compatible with multiple languages including PHP and JavaScript. It also can be integrated with GitHub making it easy to check metrics as the project progresses. In addition to its thousands of rules it also offers feedback on security and performance. Codacy is also highly secure making it viable for projects like this where data may be sensitive.

Software quality is important for the website because we will not be maintaining it in the future. Employees of the company will need to take what we have built and understand it well enough to make changes. While code standards and metrics will be used while writing it a software tool could greatly increase our successfulness in meeting this requirement. We will be implementing Codacy into our final solution. This software provides endless rules to check our final code with. It also will not reduce productivity because it can be integrated straight into github. The best part about Codacy is its professional quality and ability to catch errors from start to finish.

4.8 Database Management

Structured Query Language, SQL, is a standard computer language for database management. This is compatible with the current standards of the EEC. SQL is used to create queries which will insert, update and modify data. SQL is one of the most wide-used database management systems and has the added benefit of being supported across several different platforms. SQL has a lot of advantages and very few disadvantages. One advantage to SQL is that it is high speed and able to retrieve large amounts of data efficiently. This means that the EEC will be able to rely on this database to perform under high stress. The one drawback is that SQL is a commonly known language, so it is extremely susceptible to injection attacks. It also costs money to have an SQL server. We will model our tables to be similar to theirs and try to achieve maximum cross-over between databases. We will have several tables to manage tasks, projects, employees, hours etc.

For the database type, going with SQL is the best. Because of its wide-use, the EEC already has an SQL server implemented. Considering there are no big drawbacks to using SQL, it seems unnecessary to force a company to learn something new. Also with SQL you can access your tables through MySQL, so you do not need to know how to query to select results or view database entries. Plus, if you do need to create a query, by using MySQL it will create the query for you and give you the code to use in your webpage. Overall, SQL will provide the best service and overall best experience for the EEC.

4.9 Time Clock

The site needs a system that allows employees to clock in and clock out for certain projects, as long as they don't go over the amount of time allotted to that project. With a database, when a project is posted, it will log the start date and the amount of available hours on the task. When they assign hours to a task, the employee may go over the number of allotted hours, but they will be notified to check their manager. The hours will also be displayed in red. With this system it will insure that no one over works a certain project and that the company won't be spending extra time on projects. We decided against emailing the manager to reduce the amount of spam.

Again, the best way to handle this system is through database handling. The named time clocks are great for keeping track of when someone but not for when someone goes over hours on a certain project. With a database, we can store when a project is started and how much time it can be worked on. After doing so, when an employee goes to log hours, it will show how many hours they were assigned to the task and how many they have already logged. Once submitted, if the user has gone over the amount of hours they were assigned, they will receive a warning to contact the project manager as soon as possible.

4.10 Employee Records

The site needs to be able to keep track of employees. The page needs to include a list of all active employees and link to pages that contain specific employee information. This employee information contains name, major, the date they started at the center as well as other information. From these pages, you must be able to view and edit employee information. A database is an organized collection of data. The data is stored and accessed through tables and queries. By having a database, the EEC will be able to make changes to employee records by using their own webpage. To do so, they will need to use web forms and Get and Post requests. These requests will be able to receive and send data to and from their database securely and as frequently with no added cost to the company. There is also many different varieties of databases available to better service the needs of the EEC. Although, databases that are on personal websites are more susceptible to attacks so employees that manage the web page will have to know about security.

Employee records are private and need to be accessed on the daily, so having your own database ensures no one else has access to these records and will always be available. By using a web page and forms to submit changes to your database, you are in control of security and can be used at your disposal.

5 Design Viewpoints

5.1 Introduction

The following viewpoints and design elements will be discussed in the following format:

1) Interaction (5.2)
   - Time Clock

- Login
- Employee Records
- Projects and Tasks
- User Interfaces
- Viewing User's Work Hours

2) Dependency (5.3)
   - Security
   - Login
   - Database Management
3) Functional (5.4)
   - Security
   - Login
   - User Interfaces
   - Projects and Tasks
   - Viewing User's Work Hours
   - Code Readability
   - Database Management
   - Time Clock
   - Employee Records
4) Information (5.5)
   - User Interfaces
   - Database Management
5) Development (5.6)
   - Code Readability
   - Security
   - Database Management

5.2 Interactions Viewpoint

The interactions viewpoint will talk about the different interactions the user has with different areas of our project. These interactions include viewing or making actual interactions with the web page.

5.2.1 Design Concerns

An easy and visually appealing experience for the user is a high priority for this project. The user should be able to interact with different areas of the web page and get a quick response from the web page. A concern here is knowing the cut off between feedback and usability and how the user will interact with different features. If a feature is cost effective and may take longer to display the information, we may need to substitute how it looks for response time.

5.2.2 Design Elements

1) Time Clock
2) Login
3) Employee Records
4) Projects and Tasks
5) User Interfaces
6) Viewing User's Work Hours

5.2.3 Interaction Attributes

The user must be able to log hours into projects they have worked on. Also, users should be notified if they have exceeded the allotted amount of time for that project. To do this, the user will have the option to enter their project hours by interacting with a web page. From there, the webpage will send a request to the database and the database will send a notification back. This notification will either tell the user that everything was submitted successfully, exceeded the allotted time, or if there was an error.

When landing on the Energy Efficiency Center's web page, you should always land on the Login page. The login page will ask the user for their username and password before letting them proceed any further. Upon hitting submit, the web page will use php to check for potential security risks and send the (safe) information securely to the database and check to make sure it is valid using SQL. If the login information is invalid, a notification will be sent back to the user saying so and will not let them move any further into the webpage. If the login information is valid, the database will send a notification back to the webpage and then the webpage will redirect the user to the homepage.

The employee records page should act as a profile page. This page will display the user's personal information such as their name, contact information, start date, and projects worked on. Upon landing on this page, the web page will use php to send an SQL query to the database to retrieve the user's information. The database will then send this information back to the web page and will lay it out for the user to interact with. At the top of the page the user will see their name and personal information and have the option to edit their contact information if needed. The user will also be able to view their most recent projects they have worked on, as well as have the option to view all their projects.

The projects page should be the landing page that the user ends up on when trying to view all the available projects. From there the user will be able to click on any of the projects and be brought to a new page that displays the Gantt chart as well as a list of all the tasks for this project. When clicking on a task, it will show the description of the task, allotted hours for this task, and how many hours have been logged on this task.

On every web page, the user will have some sort of interaction with the user interface. Whether it be viewing or navigating, it should be an easy and visually appealing process.

The user will be able to view their work hours. These work hours should should users which tasks they have worked on and for how long. It should also show the user the total number of hours they have worked.

5.3 Dependency Viewpoint

The dependency viewpoint specifies the relationship of interconnection and access among entities. These relationships include shared information, order of execution or parameterization of interfaces.

5.3.1 Design Concerns

Security will be a main priority to keep the integrity of sensitive client data. To ensure the integrity and confidentiality, authenticity of users must also be verified. A secure login is the best way to do is.

The user interface determines how the employee records, assign hours, and project and task management layout will be. For example, the current project management system is extremely difficult to navigate and update. To ensure the optimal utility, the database management will also dictate functionality and layout of the requested client features. In the current website, employees may update their information in one page, but the information in the database will not update in another section of the webpage.

5.3.2 Design Elements

1) Security
2) Login
3) User Interfaces
4) Database Management
5) Employee Records
6) Projects and Tasks
7) Time Clock
8) Code Readability

5.3.3 Dependencies Attributes

Security keeps the integrity of sensitive client data. To ensure the integrity and confidentiality, authenticity of users must also be verified. A secure login is the best way to do is. Therefore, login is dependent on security.

User Interface and Database Management determines how the employee records, assign hours, and project and task management layout will be. By focusing on user interface, we can ensure the client can easily navigate the new system. In the current website, employees may update their information in one page, but the information in the database will not update in another section of the webpage.

5.4 Functional Viewpoint

The functional viewpoint describes the system's functional elements, their responsibilities, interfaces, and primary interactions. As well as focuses on the system as a whole and it's ability to change, secured and runtime performance.

5.4.1 Design Concerns

To make everything functional, it is a concern that we will be able to create an experience for users that is both visually appealing and have reliable performance. Everything should function properly and the user's view should not look cluttered or unappealing.

5.4.2 Design Elements

1) Security
2) Login
3) User Interfaces
4) Projects and Tasks
5) Viewing User's Work Hours
6) Code Readability
7) Database Management
8) Time Clock
9) Employee Records

5.4.3 Functional Attributes

Making sure the system as whole is secure is crucial to the overall functionality of the system. If the system is not secure, it will susceptibil to attacks that could render the system useless. To defend against this, we will make sure all features are secure and only authorized users can access the system.

Alongside security, having a login page be the initial landing page of the system will help make sure unauthorized users do not have access to the system. By making sure only authorized users can access the website, it makes sure the databases are more protected against attacks and private information cannot be reached.

By making a user interface that is responsive and reliable, it ensures the system as a whole is functioning at its best. Also since the user interface is how all interactions between the system and the user are handled, it is crucial that it is always functioning at its peak.

One of the features of the system is to be able to view and edit projects and tasks. To handle this, we will have a page that shows a list of all the projects and edit projects. To get this information, the page will be connected to a database with all the needed information. When the user clicks on a project, it will show all the tasks that are in the project and give the option to edit the project.

Another feature is for users to be able to clock in hours to specific tasks they have worked on. To do this, when searching for projects and tasks, upon finding and clicking on a specific project, they will be able to enter hours worked on different tasks. These inputs are then sent to a database securely so they can be reached again at a later time.

The user should also be able to view their worked hours. To see their hours, users will navigate to a page that lists the number of hours worked on all the tasks they have worked on. To do this, when a user logs their hours on a certain task, it will be stored with their specific user id. So when retrieving the tasks, we will search for the specific user id and get the needed information. By doing it this way, it ensures that all hours will be accounted for and the user will always have access to their hours.

To make sure the system as a whole can be easily changed, we will make sure the code is well commented and easy to follow so the code is easy to read. By doing this, the clients will be easily able to make changes at their discretion without having to know what all the code means.

Since most of the functionality is done through the database, choosing a database that is reliable and usable is of highest importance. By going with an SQL database, it will make it easy to manage, retrieve and store information. The database will be primarily accessed by the user through other functionality on the web page.

The final feature is the ability for user to look at their employee record. Upon clicking on the employee record option, it should show the user's name and contact information and the ability to change it, if needed. It will also show their most recent tasks they have clocked in on and the ability to check the rest. All this information will be retrieved from the SQL database.

5.5 Information Viewpoint

The information viewpoint describes the way that the architecture stores, manipulates, manages, and distributes information. The ultimate purpose of the system as a whole is to show the information in both a responsive and readable manner.

5.5.1 Design Concerns

A design concern we have is to be able to store and manipulate user data and also to show the user their information. To store the data we will be using an SQL database so the user can access their data, edit it and store it again in an efficient manner. To allow the user to access this database, we will create a user interface that is both functional but visually appealing as well.

5.5.2 Design Elements

1) User Interfaces
2) Database Management

5.5.3 Information Attributes

The database should be able to store the information in a way that makes the information retrievable and manipulatable. To best handle this, we will use a SQL database. An SQL database would make it so the information can be stored reliably and and be manipulated at the user's convenience.

To make the information from the SQL database readable, we will create a user interface. This user interface will make it so the information is both received quickly after a request is made and distributed in a way that is visually appealing to the user. The user interface will also make it so the user can manipulate the information in the database easily and securely.

5.6 Development Viewpoint

The development viewpoint describes the architecture that supports the software development process. Development views communicate the aspects of the architecture of interest to those stakeholders involved in building, testing, maintaining, and enhancing the system.

5.6.1 Design Concerns

A development concern that we have ie being able to create a system that is both secure, maintainable and responsive. To make sure the system is secure and maintainable, we will make sure the code is well commented and security is of highest priority.

5.6.2 Design Elements

1) Code Readability
2) Security

5.6.3 Development Attributes

To make sure the system is maintainable and secure, we will make sure the code readability and security are of highest priority. For code readability, we will make sure the code is easy to follow and well commented so the client will be able to make changes and updates seamlessly and efficiently.

For the security portion, we will defend against common web security attacks. To help with this, we will use php to defend against possible database attacks and to check for unauthorized access to make sure information maintains safe.

6 SIGNATURES

Anya Lehman	Date
Sr Kanna	Date
Garrett Amidon	Date
James O’Neal	Date