Web Scheduler

|  | Wilson Ho  McEwan University (CMPT396)  Edmonton, Canada  how29@mymacewan.ca |  |
| --- | --- | --- |

# ABSTRACT

This paper will discus the purpose of the Web Scheduler website and its design choices. What this application is intended to do and help users become more productive by using this website. Design choices such as database, the user interface and data visualization. This will also incorporate key design concepts that were applied to each component of creating the Web Scheduler.

## Author Keywords

Scheduler; entity relationship diagram; visualization; web application; user experience; user interface; CRUD.

# INTRODUCTION

The Web Scheduler is intended to help small working units in industries organize and manage their group’s work hours. It was built with many managers and many groups in mind. The inspiration behind this project was how I’ve noticed how extremely frustrating it can be when you’re managing a group of people and delegating work hours to run an operation. Current conventional calendar applications do not have the specific requirements needed to tailor towards user login and calendar event assignments. If you do wish to take scheduling online you usually have to pay a subscription fee. Websites such as Full Calendar offer this service from between $90 - $3200 depending on the scale of the operation and how much support you require for you organization [1].

The Web Scheduler was designed with smaller companies in mind, providing them a digital scheduling outlet, which gets the job done without paying a large amounts of subscription fees. This means allowing multiple groups and users to share the same database and accessing scheduling information pertaining only to their own group. It should allow sharing of scheduling information within groups but not between them. By doing so, one web hosting subscription can pay for many stores and many users without each individually paying $90 - $3200 a year but instead it would take $3 - $13 dollars a month to host many different organizations and groups [2]. This will help small business owners cut down on expenses if they are already using a scheduler or give other business an affordable opportunity to bring their scheduling online.

## Test Group

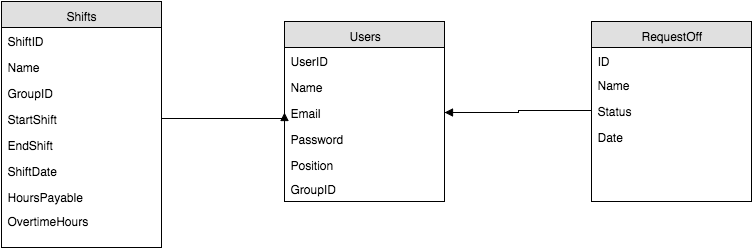
To implement the Web Scheduler I worked closely with a clothing retailor called *Street*. They are brand name clothing store associated with the Canadian clothing chain, *Below the Belt*. The store was a good fit for building up the foundations of the Web Site and launching a beta version of the Web Scheduler. Staff size around 10-15 with scheduling issues that could be resolved if they went digital. Currently the beta is heavily tailored towards their store needs, but the basic concepts and principles of developing the web application can be implemented and tested.

# Breif description of Client Speicifcations

Though an interview with the *Street* manager and an on-site observation, certain specifications were requested and interpreted for the Web Scheduler. For the project the website must be able to allow a manager to login and perform various CRUD based tasks for managing employee profiles and scheduling. It should also allow multiple users to log in with reduced functionality and view their profile and work hours. It must provide a method of planning (assigning employee with date and times of work). It should allow for updates and changes to both employees and their respective work hours. The data should be displayed in web-based visualization that is both easy to understand and displays key elements of interest. These are elements such as; who is working; when do they start; when are they done; how much time in total did they work? It should also calculate all the hours payable from one start date to another for each individual employee. An additional functionality is allowing users to request a day off from being scheduled, then allowing the manager to either accept or decline the request, so that during planning managers are reminded that they had approved a request for days off and should not assign work to that employee.

# Database Design

Given the specifications for building the website, the back end should support 3 different CRUD based actions for users, shifts, and requests day off.



**Figure 1. ER Diagram for the Web Scheduler**

The ER diagram describes how a user may have many shifts assigned to them or makes many requests for taking days off. Therefore the users database has a one to many relationships with both shifts and request off. For simplicity the current database uses the user name as a foreign key to shifts and database and places a unique constraint upon the creation of a user name. However, if the system were to be implemented on a large scale than just a single store the foreign key needs to be replaced by UserID instead.

# User tasks

The Web Scheduler focuses on two main user tasks, which are to analyze, browse, lookup and query information pertaining to the staff and their shifts.

## Present via Analyze

A shift is stored in the database with important fields of interest such as employee’s name, start and end of shift. The Web Scheduler has to be able to pull this information and present these attributes in a way that is meaningful and easy to read. The information must be organized and effectively communicated to the user.

## Browse and Lookup via Search

Users should be allowed to lookup their past shifts in the database. They may require a method to check their pay or any discrepancies that had been paid out. The web application would provide evidence based on the information recorded by the system.

Users also need to browse future upcoming shifts once available. Employees need to be able to look a head and plan for work and when the manager has assigned the shifts.

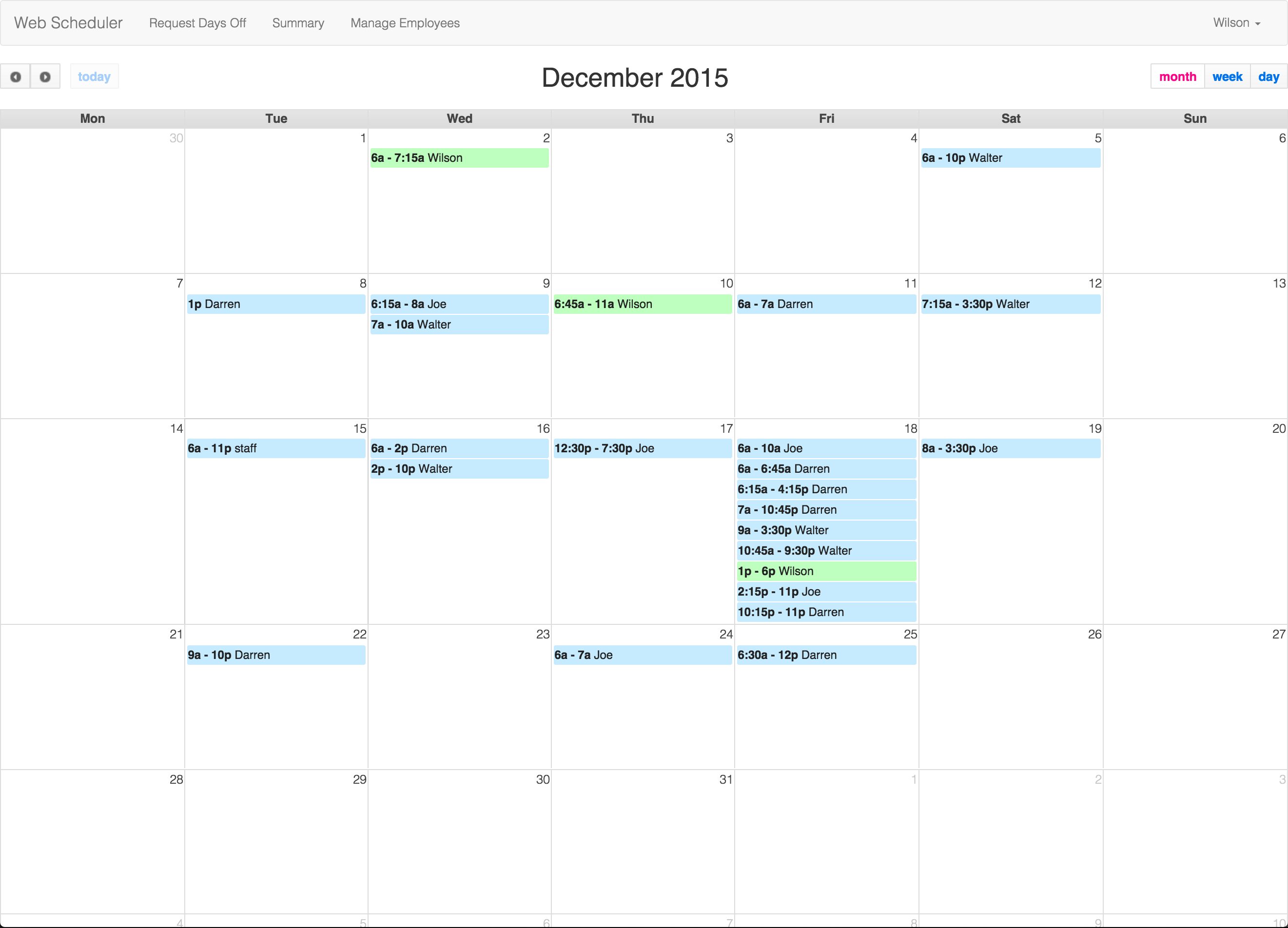
## Summarize via Query

This functionality is mainly for the managers, who should be able to summarize the shifts and pertaining to their employees. It must summarize a lookup between two dates and total up all the hours there are required to payout for an employee payroll.

# Visually encoding the data

The Web Scheduler mainly presents information from the shifts database through the open source software, Full Calendar. Listing and organizing all the shifts belonging to the group the user belongs to in a table format such as a calendar. The summary of shifts, list of requests for days off and the lookup for users are display using a table format.

## Space Filling Calendar



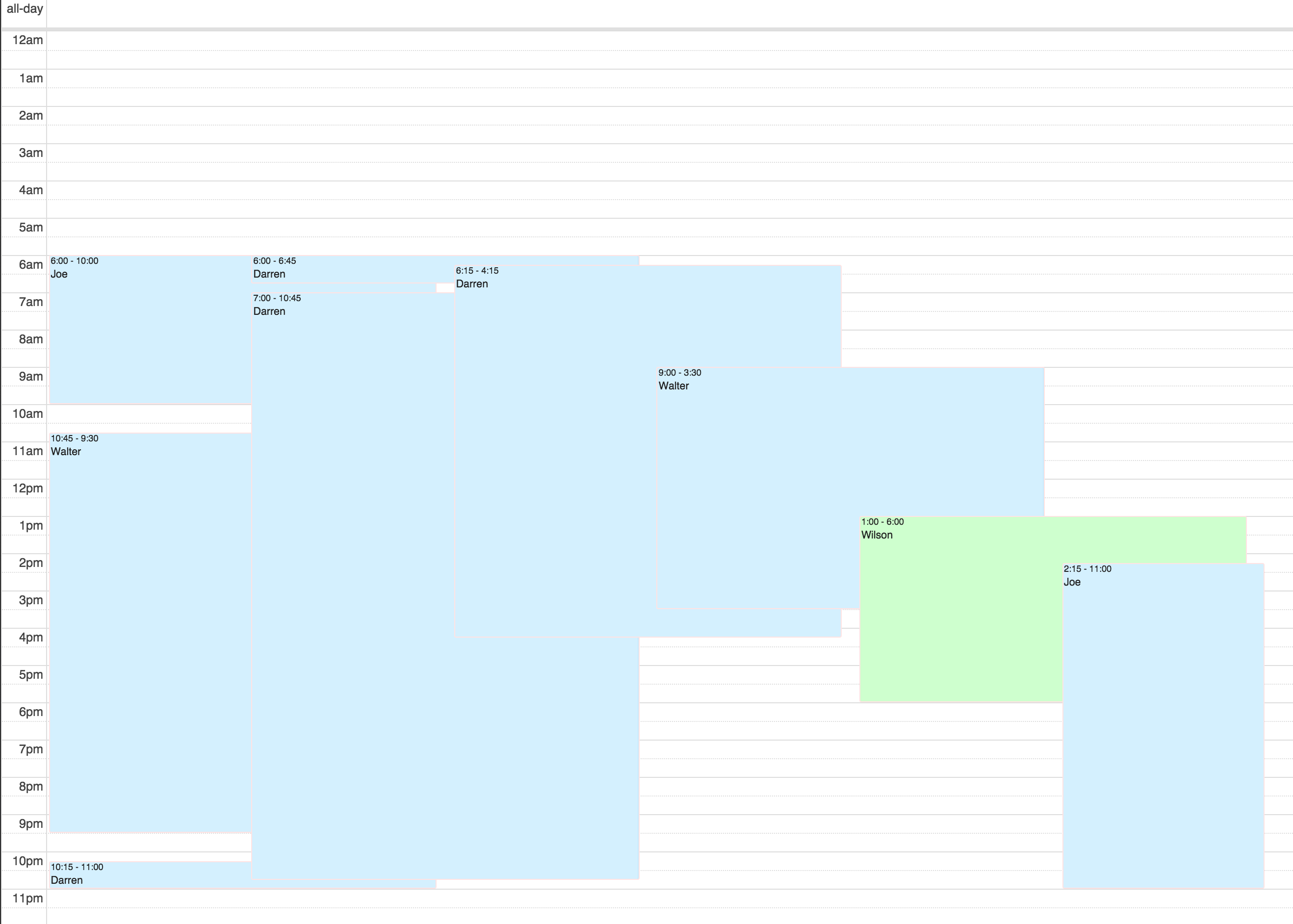
**Figure 2. Full Calendar presented to fill entire space of browser**

The calendar fills all available space in the view aside from the navigation bar at the top. The calendar format provides area marks for the shifts and their appropriate dates. This maximized the amount of room available for color-coding [3]. Color-coding such as the blue for all shifts that are assigned to the group of employees the individual is associated with and the bright green, which corresponds to the individuals, assigned shifts.

Using bright green in contrast with the blue creates a popout effect [3], making the shift of the use standout amongst their fellow coworkers. This concept follows effectiveness principle; the most important attributes should be encoded with the most effective channels in order to be most noticeable [3]. As this application mainly supports the look up task for an employee, visually encoding the shifts they are most interested in seeing increases the effectiveness of the calendar display.

By displaying shift information in such manner, the website uses strong identity channels such as spatial region and color hues to display the individuals shifts and the shifts of their group.

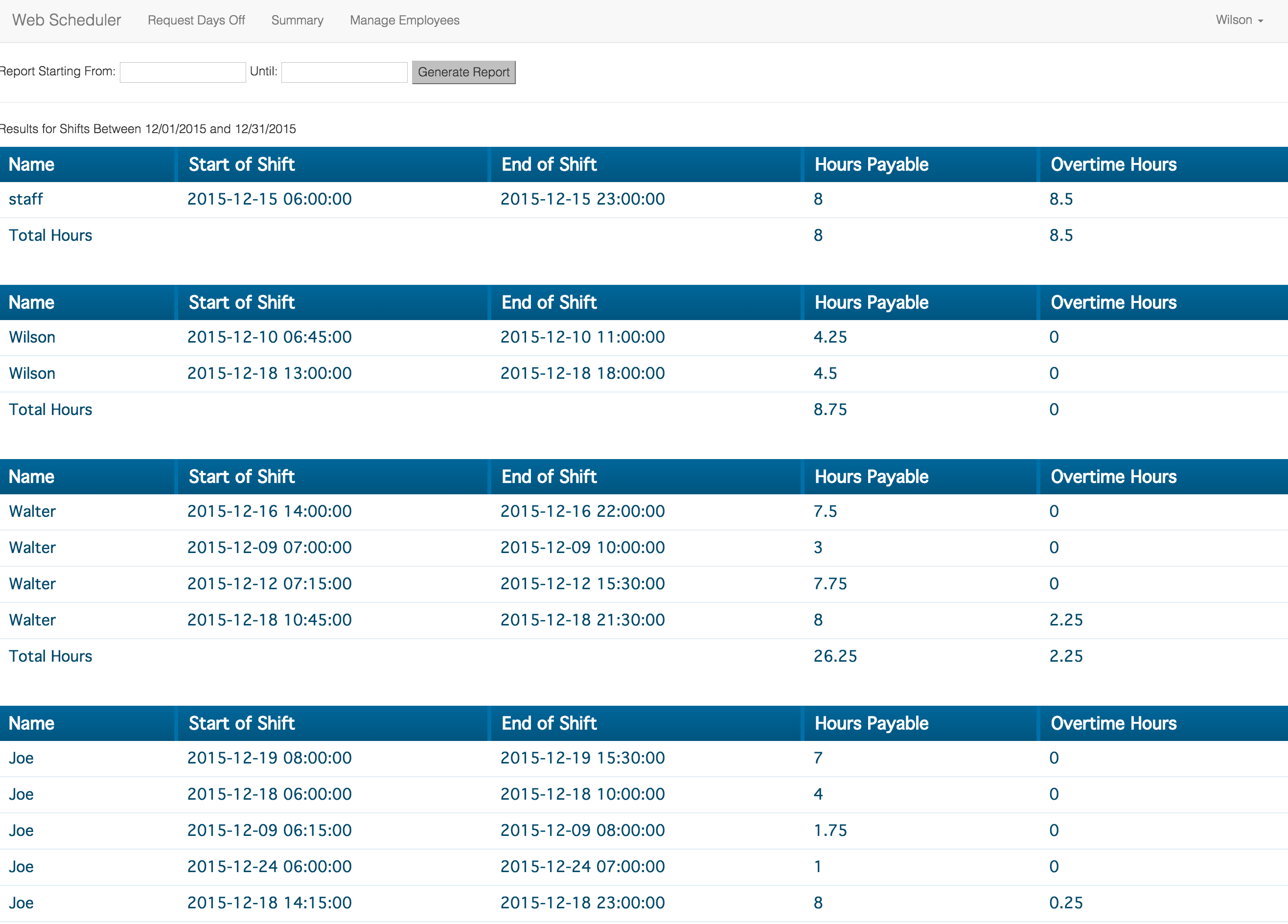
## Overview First, Details on Demand



**Figure 3. Display a detail view of shift for a particular date**

The calendar’s default view is set to monthly to provide the users with overview of all the shifts during the month following the “overview first” principle [3]. However we can visually display the attributes in the heading overview with spatial regions in the detailed view when we are looking at a particular day or week. The spatial region in figure for encodes the magnitude of the hours payable. It also aligns start of shift and end of shift with the timeline on the left.

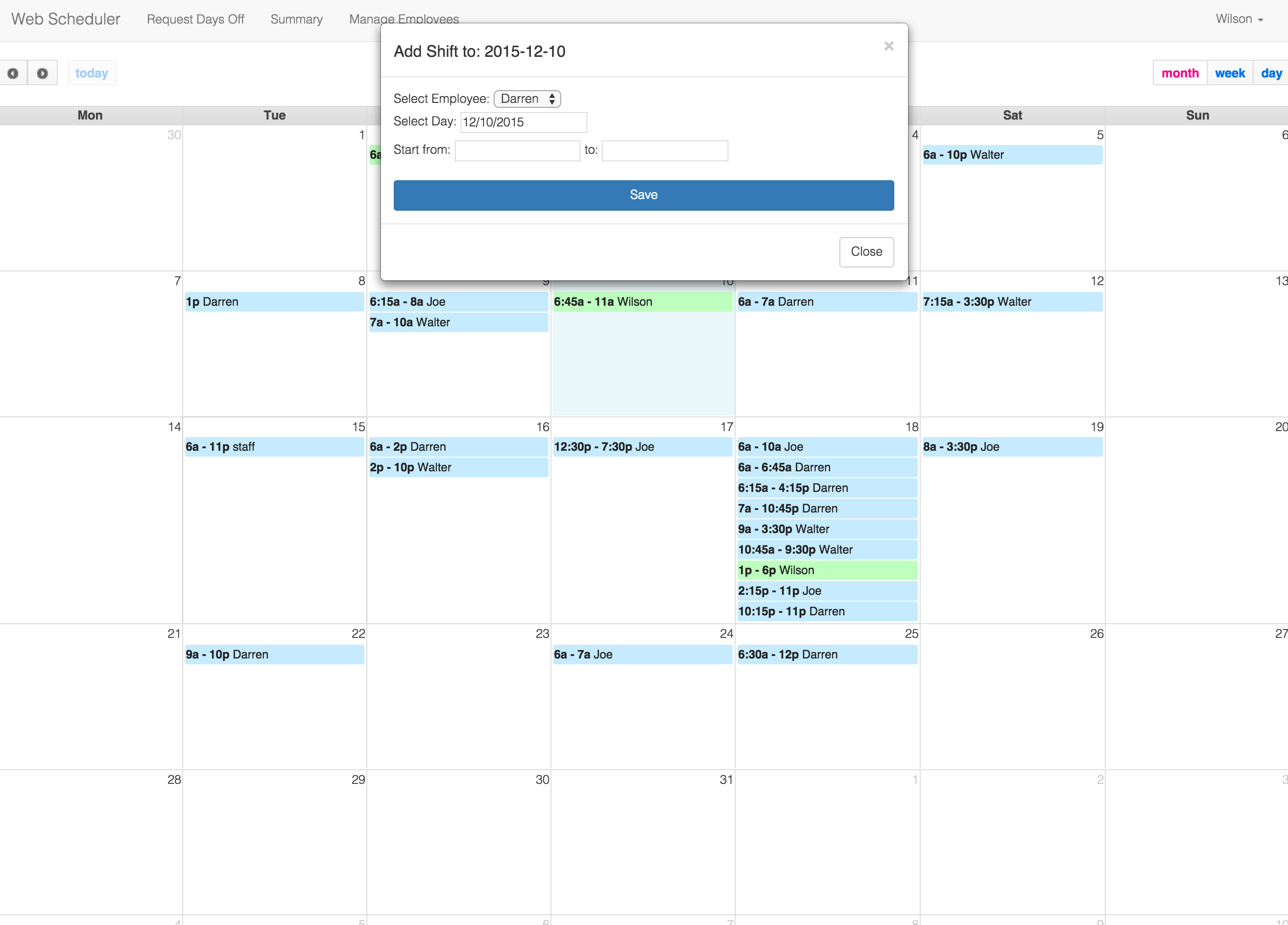
## Table Summaries



**Figure 4. Summary of all shifts within a range of dates**

There is little visual encoding provided by the summaries tables. It simply list all the shifts grouped by an employee name and sums up their total hours worked and the total overtime hours they have accumulated. There is no need to visually display this information graphically because the manager simply requires a quick look up and summed totals to appear for payroll. Having a break down of the calculation helps provide validity to the summary and create confidence when summiting the payroll. This is the same reasoning for the users table, which is organized alphabetically to do a simple lookup for a CRUD task.

## Juxtapose View



**Figure 5. A juxtapose form for adding a shift**

The concept of making a juxtapose form is that we want to keep in mind the user assign the shift would like a side by side comparison between the form and the calendar. When assigning or changing a shift a manager must keep in mind who else is working that day or how many times that person has worked that week, simply glancing between the views instead of switching between screens supports this. The calendar is permanently visible so that users can have a glance at the overview when filling out the form [3].

The juxtapose form is then incorporated in all the other CRUD tasks to keep implementation of forms consistent throughout the website. Such as adding and changing employee information or accept or declining a request for days off.

# Features yet to be implemented

There were two less significant features yet to be implemented. One being the notification system, to be able to notify all users that the upcoming shifts has been finalized. Ideally a manager can select between certain dates and submit a request to email their employees with the contents of their shift information.

The second feature is a running count of the total hours assigned for the month. Ideally this should be presented along with the calendar overview, however this could be currently done in the summaries page. Keeping a running count of the total hours assigned.

These features were left out due to time constraints and its significance. However, these are useful features to be implemented and refined if the website continues to be used.

# Conclusion

The Web Scheduler effectively handles most tasks in a very simplistic and visually appealing manner much different to its paper trail predecessor. It performs the same tasks but provides better communication, visualization and automation than using a paper trail.

Hopefully with further testing through implementing a new scheduling system with *Street*, the Web Scheduler can be adopted by all stores in the *Below the Belt* clothing chain and grow to provide the stores with a modern, effective way to communicate scheduling information with an alternative cost effective website.

# REFERENCES

1. Adam Shaw. (2015). *Scheduler.* Retrieved from Full Calendar website December 3, 2015: <http://fullcalendar.io/scheduler/purchase/>
2. HostPapa. (n.d). Retrieved December 3, 2015, from [http://www.hostpapa.com/](http://www.hostpapa.com/lp/hosting-review?transaction_id=1020b692a0f29991b5ba35c2fd7ff6&gclid=&utm_source=hosting-review&utm_medium=organic&utm_campaign=US-EN&utm_term=&utm_content=&transaction_id=1020b692a0f29991b5ba35c2fd7ff6)
3. Tamara Munzner. 2014. *Visualization Analysis & Design*. CRC Press.