Book-It

Christopher Compton
Jacob Nappi
Thu Tran

91.462 GUI Programming II
Professor Jesse Heines
Project Proposal

February 04, 2014

Contents

Goal Statement

The goal of our project is to provide a web application for small business owners with an emphasis on 1-to-1 interactions that will allow their clients to book appointments with their employees in a fast, easy-to-use way.

Feature Descriptions

Appointment Calendar

The appointment calendar is the focal point of our application. Users will be able to log on and select with whom they would like to make an appointment. The respective appointment calendar will then be displayed to the user.

Different Views: Month and Day

- The calendar will first be presented to the user displaying the current month with the current date highlighted. The user will be able to switch months as needed in order to find the date they wish to make an appointment. In this view, the calendar will differentiate between days that are fully booked and days that have available appointment slots.
- Once the user selects a day that has an available slot, they will be presented with a more detailed view of that day. Just as in the month view, the day view will differentiate between free and booked time slots.

User Login

We will have users create accounts and login in order to use the application. This has a number of benefits, both to the users and to us, as the developers.

Purposes

- Customers will have their personal information stored to automatically fill in appointment forms. The goal of this is to make it faster and easier for the user. The user can still edit the form as needed, if for example, a parent is making an appointment for a child.

- Both customers and employees will be able to manage their calendars. Customers will have access to only their existing appointments and will be able to add, modify, or cancel appointments as needed. Employees will have unrestricted access to all of their client appointments. They will need to manage their entire client base in the case that they have to cancel or move an entire day's worth of appointments.

- Lastly, this will make development easier on our part. It will be easier to develop solely for registered users than for both registered and guest users.

Appointment Form

This is how the user will create their appointment. It will contain fields for personal information. This form will have an idle timer that will return the user to the home page when they are idle for a set amount of time.

Feature Components

Calendar

The calendar will be based in HTML 5 and we will populate the calendar using information that will be stored in a MySQL database. To build the HTML 5 table and dynamically fill it we are going to use JavaScript. To accomplish this, we will reference the following link:

http://www.code-sucks.com/code/javascript/template.php?tutorial=basic_calendar.php [1]

User Login

The user login feature is going to be based in PHP and MySQL. We will use a PHP wrapper to communicate with the MySQL server where the user's credentials and personal information are going to be stored. This will allow the user to only view their appointment information on the calendar. They will not be able to see the personal information of other users. To accomplish this, we will reference the following link:

http://net.tutsplus.com/articles/news/how-to-build-a-login-system-for-a-simple-website/ [5]

Appointment Form

The appointment form will have an idle time out which will return the user to the homepage. This is accomplished by setting JavaScript event listeners to track key strokes and mouse movements. Events caught by the event listeners will reset the timer.

Mutual exclusion for appointment modification

Because of the possibility of multiple people attempting to add an appointment for the same time on the calendar we need to have a system in place to queue up the users. For this we plan to use a code methodology called Comet which allows the server to ping the client and let it know when data entry is allowed. To accomplish this, we will reference the following link:

http://ajaxian.com/archives/comet-a-new-approach-to-ajax-applications [2]

MySQL backend to store calendar information

In order for the user to be able to enter in an appointment and have that information persist, we are going to use a MySQL server with a PHP wrapper to facilitate communications with the MySQL server. The MySQL database will store a table of users and a table containing appointment information for each month. To accomplish this, we will reference the following links:

http://www.w3schools.com/php/php_mysql_intro.asp [4]

http://www.php.net/manual/en/book.mysql.php [3]

User Descriptions

Our application will be focused on two user bases; the business customers and the business employees.

Customer

The customer is the focal point of this application. It is designed specifically for their use in mind. As such, the application needs to be extremely simple to use such that anyone that is able to navigate a web browser should be able to use this application to create and manage their accounts and appointments.

Employee

Similar to the customer, the employee needs to be able to use this application easily. Although the goal of the application is to reduce the amount of time the employee spends adding, modifying, and canceling appointments, the need for the employee to manage their calendars cannot be totally eliminated. Ease of use for the employee will be on par with that of the customer. Anyone that is able to operate a web browser should be able to use our application.

Potential Issues

- Failure to implement the Comet methodology for queuing users
    - On user submit indicate to the user success or failure of creating an appointment

- Having employee user functionality implemented in time
    - This is something that is time permitting because our primary focus is customer functionality

Schedule

		Compton	Nappi	Tran
9-Feb-14	Set up Git repository. Develop home page, Calendar Page, and Form using HTML, JavaScript, and jQuery UI			
16-Feb-14	Style pages with CSS. Save Form data to session. Refresh calendar with session saved data.			
23-Feb-14	Alpha complete by this date. Begin Beta development.			
2-Mar-14	Design and create calendar database			
9-Mar-14	PHP API for database communication			
16-Mar-14	Implement User Login and Accounts database			
23-Mar-14	Implement queueing system for users			
30-Mar-14	Implement employee login			
6-Apr-14	If time permits, add the "nice to have features"			
13-Apr-14	Polish up Beta and prepare for usability testing.			
20-Apr-14	Completion of Beta Version and Usability Testing			
27-Apr-14	Bug Fixing and polishing.			
2-May-14	Completion of Project for Presentations			

Acceptability Criteria

Minimum Required Features

Our main goal is to be able to have the customers log in and make appointments. This means that the customers need to be able to create an account, log in, create new appointments, and modify or delete existing appointments.

Time Permitting Features

If time permits, we will implement the employee accounts to allow employees to manage their respective calendars similar to the way customers can manage their appointments.

Future Features

Allow for customers to select from services provided, i.e. hair coloring vs. a trim. This would require that time slots be dynamically assigned to the appointments since different services would require longer or shorter appointments.

Works Cited

1. code-sucks. (n.d.). *Basic JavaScript Calender*. Retrieved February 02, 2014, from code-sucks.com: http://www.code-sucks.com/code/javascript/template.php?tutorial=basic_calendar.php

2. Mahemoff, M. (2006, March 04). *Comet: A New Approach to Ajax Applications*. Retrieved February 02, 2014, from ajaxian.com: http://ajaxian.com/archives/comet-a-new-approach-to-ajax-applications

3. The PHP Group. (n.d.). *Original MySWL API*. Retrieved February 02, 2014, from php.net: http://www.php.net/manual/en/book.mysql.php

4. w3schools. (n.d.). *PHP MySQL Introduction*. Retrieved February 02, 2014, from w3schools.com: http://www.w3schools.com/php/php_mysql_intro.asp

5. Way, J. (2009, January 29). *How to Build a Login System for a Simple Website*. Retrieved February 02, 2014, from net.tutsplus.com: http://net.tutsplus.com/articles/news/how-to-build-a-login-system-for-a-simple-website/