2/21/2018

XSchedule Software Requirements Document

Team Members: Sonia Azad, Timothy Ratliff, Joseph Nguyen, Mark Prutz, Dylan Russell

Version 1.0

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to present a formal documentation of our web based application XSchedule (Version 1.0). This document will highlight the functions and features, technical specifications, and web functionality, as well as the application's purpose.

1.2. Intended Audience

The following document is intended for technicians and managers to grasp a full understanding of the application and the functions available. This document will inform readers of the technical specifications, features, website functionality, and other traits that XSchedule has to offer.

1.3 Product Scope

XSchedule is a web based application scheduling service for MODEL Computing Services. The application will allow customers to request a work order in real time. Customers of MODEL receive priority based on how many services they received from MODEL in the past. Technicians working for MODEL will accept work orders, from the queue, that the customers request. Technicians have the ability to accept work orders and document hours until completion. The system will then generate a bill for the customers based on working hours and technician experience. Managers have all the abilities of a technician in addition to extra abilities. Managers can view, access, and modify data about: the queue, work orders not accepted, worker performance, and times technicians are left idle. The application will be named XSchedule referencing fast scheduling services that their customers desire.

1.4 User Interface (UI)

The user interface will be different between customers, technicians and managers. Customers will be given the option to post a work order on XSchedule and receive an estimated quote on how much their request may cost. Technicians will be able to see the queue of all the work orders and accept them. Once an order is accepted, it will be displayed on the top of the web page stating "in progress." Technicians will also have work ID's and years of experience listed. Managers will have information about the queue displayed on their web page. This includes: average queue length, individual technician idle times, and time the queue is empty in the form of a (*whatever we are using here*) graph. Summary reports for the month will be displayed as well.

2. Technical Specifications

2.1 Team Communication

Our team will use the communication app Discord as the primary method of tracking deadlines and meetings. As specified in the project requirements, all code will be available on GitHub, and each member will be responsible for keeping their own contributions to the project updated through frequent commits to the GitHub repository.

2.2 Tools and Development

Our team will use Microsoft's Visual Studio as the primary IDE for coding. The project will use an ASP.NET Web Forms template and function as a live website, running on Amazon Web Services Elastic Beanstalk.

2.3 Amazon Web Services

As previously mentioned, XSchedule will run on AWS Elastic Beanstalk for maximum flexibility and scalability. To minimize compatibility issues, our servers will be of the following specification: *AWS t2.micro 64bit Windows Server 2016 v1.2.0 IIS 10.0*

3. Features

3.1 Users

There will be three types of users on XSchedule: Customers, Employees, and Managers.

3.1.1 Customers

Those who use the application to request work orders. Customers will have the following information stored in their profile:

- ID number
- Name

- Number of times serviced

Customers can create a work request with the following information:

- Name
- Difficulty level (1-3)
- Location
- ID number

3.1.2 Employees

Employees can see a list/queue of work requests. Work requests have the following information:

- Customer priority
- Complexity
- Time stamp

Employees can set a work request as active and mark when a task is completed.

3.1.3 Managers

Managers can view all website data, including Customer and Employee profiles, the work request queue, and the time it took for an employee to complete a task.

3.2 Product Functions

- Login/Profile
- Real time work order estimates
- Real time queue
- Notification for completed work orders
- Acceptance/Removal of work orders
- In-progress notification bar
- Data Information
- Work Queue Edit

3.3 Website Functionality

3.2.1 Login

All users must login with a username and password (stored in a secure database) to access the application. The login page will check the credentials entered, and if deemed correct, grant access to the application.

3.2.2 Register

In order to access XSchedule, the user must register a username and password on the register page. The register page will accept these credentials and save them to the secure database.

3.2.3 About

The about page will contain information about the project, purpose of the application, and the course (Software System Development).

3.2.4 Contact

The contact page will contain links to each group member's GitHub profile page, and a short bio about the member.

4. Functional Requirements

4.1 Queue Specifications

The following information about the queue will be recorded as management has requested:

- Average wait time before a job is started
- Average queue length
- Percentage of time the queue is empty
- Number of jobs not addressed on their request date
- Number of hours each day a technician is idle

5. Non Functional Requirements

5.1 Platform

Our application will be demoed and ran from:

- Microsoft's Visual Studio

5.2 UML Models

Primary Modeling Tool:

- UML

5.3 Group Meetings

Group Meetings are recorded and saved in a document that can be accessed at any time.

5.4 Github

Github provides an interactive experience for our team. Each member may contribute to the repository until a final product is reached. Our project github can be found here.