CS 480 – ADVANCED SOFTWARE ENGINEERING

Right On Time

SRS Requirements

Laura Leist

11/17/2016

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to present the expected requirements of the requested software as well as defining both the functional and non-functional requirements. It will serve us as a guide throughout the development process regarding the specifications and design of the software. It is also to be used as a tool to present to the clients what it is we have understood from their given specifications and descriptions.

1.2 Scope

The clients have asked for our help in producing a software that gives their employees the ability to clock in and out their hours worked on a specific task, while requesting necessary materials for their project/task. We will be producing both a website and a mobile app to provide employees access to the software both in their office and on the go around the facilities. The software will not be used for such actions as clocking in and out of work and will not provide a means of "instant communication" among coworkers.

The goal of this software is to keep track of the various projects underway at the company and be a source of project-status communication between managers and employees as to what equipment is ordered or what tasks have been completed in order to continue on with a project.

1.3 Definitions

BPT Design – Breedt Production Tooling and Design

"Right On Time" – Title of the product

BOM – Bill of Materials

Admin – Head of the BPT Design company

1.4 References

IEEE Software Engineering Standards Committee, "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications", October 20, 1998.

1.5 Overview

This rest of this document contains information and goes into further detail regarding the Right On Time software that is currently under development. It will specify the system functionalities as well as interfaces throughout the software. It will also describe the specifications and requirements posed by the client and how we are to achieve them.

2. Overall Description

2.1 Product Perspective

Right On Time's system includes both a desktop and mobile compatible website where admins, managers and employees can clock in and out of projects and make edits to existing projects. The mobile application should be limited to clocking in and out while the desktop version provides more functionality regarding requested materials, editing to-do lists and such. The application will be a tool of status communication between project members, managers, and admins in order to make sure projects are being completed on time. There will need to be a ranked permissions system in order for users of different superiority levels.

The interface for Right On Time must be adaptable to both desktop and mobile. It will be in the form of a website, but will have to be very simple and clear to use in order to cater to a spectrum of employees. To be able to perform as a website, it must properly function on all major browsers such as Google, Firefox, Explorer, Safari, etc. and should perform on all platforms (Mac, PC, I-phone, and Android).

This software is going to be very heavy in data regarding employee information, different projects and their completion times, cost restrictions, and the BOM where requested equipment can be placed for approval and have it ordered. Our decision is to use MongoDB for our database and use of the JSON object language because of the unexpected amounts and types of information that might be going into the system. Flexibility is crucial.

2.2 Project Functions

Employees will be given login information and passwords, as well as their designated project's access code that will give them access to their project information. As mentioned in the product perspective, the mobile version of the software is limited to clocking in and out of assigned projects while the desktop is capable of clocking in and out as well as various other functions. On the desktop version, the user should be able to request funds for a specific project, order materials and be notified when they are shipped, edit expected hours/days until completion, edit to-do lists, delete projects or pause them, and view their daily progress.

There is also a user level system granting users a specified allocation of permissions. The admin needs to be able to access all information regarding each project and BOM, while managers are limited to less actions. Employees are restricted from editing data completely, and are curbed only to clocking in and out of their assigned project and viewing the overall progress of that day and necessary data such as completion times.

2.3 User Characteristics

Within the software, there are various user levels: administrator, shop manager, project manager, purchasing, and employee. The administrator will have access to all features of the software and data regarding projects and employees. The admin will also have the authority to override any BOM or budgeted hours. The shop manager has the ability to edit and add all shop

project tasks. Both shop managers and project managers can request funds and approve hours assigned to each task. The purchasing team/accountant will only have access to the project view and requested BOM pages to assign funding, and employees will be limited to the project views and clock-in/out feature.

2.4 Constraints

Several posed constraints are if a person has a mobile device that can access internet, the quality of internet that the company uses, information security, and the amount of data going into the system.

This software is a web-based application, so it is unusable if connection is lost. With the software being hosted on the internet, customer and company security could be at risk. We need to ensure that there is a high level of protection within the software to prevent outside sources from accessing information, or even employees gaining permissions that they were not meant to receive. Lastly, in regards to data, we have no estimate on how much data will be contained within our databases. If the company continues to grow, will our database size constraints need to broaden more? For further maintenance we will need to take this into consideration.

2.5 Assumptions and Dependencies

For this SRS and throughout the development process, we will assume that all employees will have the technology and devices necessary in order to use the mobile version. This goes with the technology quality of the BPT Design company. We will assume that their internet capacity is large and fast enough to run this software on many devices at a single time, as well as their desktop devices being of proper dimensions to display the software correctly.

2.6 Apportioning of Requirements

None specified at this time.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interface

The user interface is dependent on what type of device that they're accessing the website through. There will be different displays for the desktop version and for the mobile. Our interface will also need to adjust dynamically in regards to the user-level permissions of the user. Some features may not be accessible to them and displays will need to adjust accordingly.

The user will begin with the login page. The only properties visible on this page should be textboxes for the username and password, as well as an option to create an account if they don't have one already. After login, the user is directed to the dashboard view. The UI for this should contain buttons to clock in/out of projects as well as having navigation tools to enter the Project View List or User Settings.

The Project View List will contain a list of projects that the user is a member of. This list will be selectable and will display further views and information of the selected project.

Lastly, a user settings page will be available for the user to access extra information or actions regarding their account and the software. This will be separate from the rest of the UI designs

3.1.2 Communication Interfaces

The software needs strong communication with its database. The software relies heavily on inputted data such as hours, employee information, BOMs, and project info. This data will need to be inputted, received and edited efficiently with no errors in order for this software to work as it should.

3.2 Functions

Name:	Log In
Purpose:	User will log into their account and access their project files.
Input/Output:	Input: Username and password Output: User will be directed to their dashboard.
Relationship:	This feature is of top importance and is related to all other features in the software. If a user cannot log in, they cannot access the rest of the software.
Data Formats:	String data and Boolean values

Name:	Verify User
Purpose:	Verifies with database as to whether or not the information submitted exists.
Input/Output:	Input: Submitted data from user. Output: Possible error message or navigation to dashboard.
Relationship:	Log In, User Settings
Data Formats:	String data and Boolean values.

Name:	Data retrieval
Purpose:	Retrieve data regarding employees registered projects, progress, and user permissions.
Input/Output:	Input: User information Output: Data regarding user's permissions and past data.
Relationship:	Dashboard, Database, Project View, User Settings
Data Formats:	Number values regarding days, time, and money as well as project information.

Name:	Fill Dashboard
Purpose:	After data retrieval, the data is displayed
Input/Output:	Input: Database retrieved info Output: Various displays to users regarding project data.
Relationship:	Project View, User Settings, Database, Dashboard
Data Formats:	Project Database data

Name:	Employee Dashboard
Purpose:	Current projects and their to-do lists show up here, as well as a "clock in/clock out" button and a "finish task" option.
Input/Output:	Input: Clock in/out, selection of project list view or project view Output: Navigation to other views or time stamps sent to database
Relationship:	Project Database, Project List View, User Settings, Project View
Data Formats:	String data and Boolean values

Name:	Dashboard Navigation
Purpose:	From the dashboard, the user can navigate to any wanted location of the software such as the project list view, project view, or settings.
Input/Output:	Input: Click event to send user to desired location. Output: User sent to location.
Relationship:	This feature is the key to transportation throughout the system.
Data Formats:	Click event

Name:	Clock In/Out
Purpose:	User can input project times from the main page and add time to their project.
Input/Output:	Input: Clock in/out, time stamps Output: Time stamps sent to database
Relationship:	This feature is of top importance and is related to all other features in the software.
Data Formats:	Number and time data

Name:	Project List View Fill
Purpose:	Project List is filled with either the employee's current projects that they can access, or all projects regarding permissions of shop managers and admins.
Input/Output:	Input: User permissions, Database employee info Output: projects available to view
Relationship:	User settings, project database
Data Formats:	String data, project database data

Name:	Project List View
Purpose:	Navigation to each project currently underway at the company or a list of project currently being worked on by an employee.
Input/Output:	Input: Click event to send user to desired location. Output: Depending on permissions and if they're assigned to a project, user will be sent to project view of selected project.
Relationship:	This feature promotes navigation to projects that the user might be searching for or if they're joining a project that they were assigned to.
Data Formats:	Click event

Name:	Project Selection
Purpose:	Selecting a project either from the project list or from the dashboard.
Input/Output:	Input: Click event to send user to desired project. Output: User sent to project.
Relationship:	Dashboard, Project List View, Project Database
Data Formats:	Click event

Name:	Project View
Purpose:	Project information such as budgeted time, BOM, tasks, and to-do lists can be accessed here.
Input/Output:	Input: Edits of data Output: Refreshed data is sent to database and displayed.
Relationship:	This relates to the project list view and dashboard and returns the needed data to all users of any level.
Data Formats:	String data, project database data

Name:	User Settings
Purpose:	User can edit personal information here as well as add or delete accounts or change their password.
Input/Output:	Input: click event to enter settings page, inputs into personal information or password resets Output: Refreshed personal information or new password stored in database.
Relationship:	Log in, employee database
Data Formats:	String inputs, employee database data

3.3 Performance Requirements

The system must dynamically adjust to the user levels upon logging into the system. The interfaces will change in regards to what permissions they are given. The same will be performed when using mobile devices versus desktop computers. All systems should take under one second to load requested data. Administrators may experience a slightly longer waiting period due to the quantity of information accessible to them, but nearly 100% of the time, performance should be that of under one second.

3.4 Logical Database Requirements

This software has the potential to contain a lot of data, and will most certainly grow in size throughout the future. Our databases are the pinpoint of our utmost concern when it comes to our speed and precision.

Database:	Projects
Information:	All project titles, budgeted hours, requested/purchased materials, group classification, and due dates are contained within this database.
Frequency:	This software is centered on keeping track of projects. The Project Database will be utilized in nearly every function of the system.
Access:	This database provides access to the Dashboard Fill and Project List Fill functions.
Entities:	projectID, title, location, dueDate, budgHours, BOM
Constraints:	All entities are not null except for BOM
Retention:	Will need to have separate long-term storage to retrieve data for fiscal seasons

Database:	Employees
Information:	Details of employees
Frequency:	Not as high as the Projects Database, but will still become busier for each new employee added on.
Access:	This database provides access to the Dashboard Fill and User Settings functions.
Entities:	empID, empName, permissions, groupID, dailyProg
Constraints:	All entities are not null
Retention:	Long term storage for permanent employees

Database:	BOM
Information:	Bill Of Materials for all projects
Frequency:	Often used and will be frequently updated as materials are ordered and then delivered.
Access:	This database provides access to the Projects View
Entities:	groupID, projectID, department, cost, budget, deliveredYesNo
Constraints:	All entities are not null
Retention:	Long term storage for permanent employees

3.5 Software System Attributes

3.5.1 Reliability

Our goal is to have 100% reliability in this software. If the system were to lose data, this could cause the company to lose track of a project and can result in upset customers. It could also prevent the company from having all their required information at the end of the fiscal year when they need to enter in the projects that they have completed and the money they spent and earned on each one, and who they did the project for.

3.5.2 Availability

Availability is dependent on access to the internet. If the company loses connection, no data can be entered and projects risk having their due dates moved back.

3.5.3 Security

Usernames and passwords must be unique for each employee and must be cryptic enough that there is relatively zero risk of someone hacking their account.

It is important that the system make logs and keep a history of the data that is stored within in case any unexpected issues arise or if data is accidentally erased. In relation, communication between different functions must be restricted to prevent anyone without permissions to access data.

The system contains sensitive information such as financial information that cannot be mishandled. This could cost the company money and push back due dates for projects. This will require skillful cryptography to prevent anyone from accessing this data.

3.5.4 Maintainability

The system's databases must be extendable enough to support any entities that might be added in the future. They must also be flexible to withstand large amounts of data that will be stored. Our team will meet frequently to structure out code that will be easy to write to in the future if any edits need to be made. Our code must be professional. Maintainability requires skill and time to design extendable structures, so our team will refuse to fall short of anything less.

3.5.5 Portability

Our system must be portable on both PC and Apple, as well as iOS and Android smartphones.