Software Requirements Specification

for

Right On Time

Prepared by Robert Roppo

All.dat

11/10/2016

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to layout a detailed description of Right on Time. It will explain the purpose and features of the software, what the software will do, and the design specifications of the software. This document is intended for both the clients and the developers of the software. This document will be proposed to the clients for approval.

1.2 Product Scope

The software being designed will be called Right on Time. The system will manage each project by keep track of costs, hours, and tasks. In addition to keeping projects on time, the system will keep track of build materials for each project. This will help the owners keep track of materials that are needed for specific projects. There will be several different user roles in the software in order to keep management of projects easy. The main goal of the software is help the company organize and keep track of projects.

1.3 References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.*

IEEE Computer Society, 1998.

2. Overall Description

2.1 Product Perspective

This is a new system is to help managers at the company keep track of current projects being worked on in the shop. The software will evolve over several versions, with the end goal to track costs, as well as managing projects. Administrators will create projects to which users are assigned to. These projects will have a specific amount of hours and materials budgeted. These hours are track by users completing tasks within a given project.

2.2 Product Functions

- Project View
- User View
- Bill of Materials (BOM) View
- Project List

2.3 User Classes and Characteristics

Due to the size and nature of the customers' business, there will many user levels with different permissions. These levels and permissions are laid out below:

- All levels will be able to view the Project Page, BOM, and budgeted hours.

Administrator:

- All access

Shop Manager:

- Edit tasks
- Edit BOM
- Edits projects

Project Manager:

- Edits BOM
- Edits projects

Purchasing:

- Limited to BOM acknowledgement of materials requested is on premises.

Employee:

- Start/End Tasks only
- Request BOM materials

2.4 Operating Environment

1. The software will operate on the following web browsers: Internet Explorer, Microsoft Edge, Mozilla Firefox, and Google Chrome.
2. The software will operate on an Amazon Web Server.
3. Software will operate on most computers and smart phones.

2.5 Design and Implementation Constraints

Since the software is a web application we will be limited by browsers. The software will need to be able to work on most modern browsers to ensure it can be used anywhere. In addition to compatibility on computers, the software will need to perform on mobile devices. This means that we will need to ensure it works on most modern browsers.

2.6 User Documentation

1. The software will be deployed with a paper tutorial on how to use the system.
2. The developer will also provide the company with a user manual.

2.7 Assumptions and Dependencies

A number of factors that may affect the requirements specified in the SRS include:

- Administrators keep track of tasks to make sure they are closed by lower level users.
- The task of BOM is maintained by the administrator.
- Users are assumed to maintain reporting of task completion.

External Interface Requirements

2.8 User Interfaces

- The software will show different UI options based on user level.
  - The admin view will be the only view that can see the full UI functionality.
- There will be a separate UI for mobile web and desktop web.
- The software will run on most modern mobile browsers.

2.9 Hardware Interfaces

Since Right on Time is a web application, the only requirement for the use of the software will be an internet connection.

2.10 Software Interfaces

- The software will communicate with a mongoDB database to store user and project info.
- The software will utilize C# to communicate between the user interface and the database.

2.11 Communications Interfaces

The software will send a notification in certain circumstances:

- When a task is not completed for the day.
- When the to-do list has been updated.

3. System Features

3.1 Create/Edit Project

3.1.1 Description and Priority

This feature creates the projects that the customer needs to keep track of. The project is also editable. The priority for this feature is high.

3.1.2 Stimulus/Response Sequences

Stimulus: User wants to create a new project.

Response: The system prompts the user to input details for project name and PO #. It then creates and object for this project and stores it in the database.

Stimulus: User wants to edit the project information.

Response: The system retrieves changed information and saves it to the current project object. This overwrites the previous information for the project.

Stimulus: The user wants to delete a project.

Response: The system will check to make sure the user is an administrator. If yes, the system will confirm the selection and delete the project from the database.

3.1.3 Functional Requirements

- Project.create: The system will allow an administrator to create a new project.
- Project.setId: The system will let the user set the project Id.
- Project.setPO: The system will let the user set the PO # for the project.
- Project.edit: The system will allow the user to edit the Id and PO fields.
- Project.isAdmin: The system will check to make sure the user is an admin. If yes, the user can have create/edit/delete permissions.

3.2 Bill of Materials (BOM)

3.2.1 Description and Priority

This feature keeps track of the Bill of Materials list for each project. These items can be created, edited, and deleted. In addition to this the list can be classified as completed or not completed. Each BOM is specific to a given project. The priority for this feature is high.

3.2.2 Stimulus/Response Sequences

Stimulus: The user wants to add a new item to the BOM list.

Response: The system prompts the user to input the item information then stores it in the database.

Stimulus: The user wants to edit an item on the BOM list.

Response: The system prompts the user to select the item to edit. The user then inputs the information and the item is updated in the database.

Stimulus: The user wants to toggle that all items are purchased and in stock.

Response: The system toggles a Boolean value based on the input. This is then displayed on the user interface.

3.2.3 Functional Requirements

- BOM.add: The system will add a new item to the BOM list for a specific project.

- BOM.edit: The system will allow the user to edit a specific item in the BOM list for a specific project.
- BOM.delete: The system will slow the user to delete an item from the BOM list within a specific project.
- BOM.id: An id number the system assigns the list to correlate to its specific project.

3.3 Tasks

3.3.1 Description and Priority

This feature keeps track of tasks, which are used to keep track of hours on a project. Tasks can be started, paused, and finished by the user. The project view will use this feature to tally total hours for a project. The priority for this feature is high.

3.3.2 Stimulus/Response Sequences

Stimulus: The user wants to start a task.

Response: The system will prompt the user input task and project number. Once a t ask is started, the system keeps track of hour many hours it has been active.

Stimulus: The user wants to pause a task.

Response: The system prompts the user for a reason for the pause. The system then takes hours tallied and adds it to the project. The task is then paused an hours reset and paused. The information on the pause is sent to the administrator.

Stimulus: The user wants to finish a task.

Response: The system closes the task and updates the hours from the task to the total hours for the project. This is the same for an un paused project.

3.3.3 Functional Requirements

- Tasks.start: The system will start a task and file it under a project Id.
- Tasks.pause: The system will pause a task. When this is done, hours are added to the total project hours. The user is then prompted to give a reason for the pause which is sent to the admin.
- Tasks.resume: Starts a paused task, treats the hours as if a new task has been started.
- Tasks.finish: The system completes a task. This add the hours of the task to the total hours for the specified project and closes the task.

3.4 To-Do List

3.4.1 Description and Priority

This feature is a list created by the administrator for employees to follow. The to-do list will only be able to be edited by administrators. This feature is a medium priority.

3.4.2 Stimulus/Response

Stimulus: Administrator wants to create an item on the to do list.

Response: The system will prompt the user to enter text for the task. The system will update the to-do list in the database and update the list. on lower level user views
Stimulus: Administrator wants to change an item.

Response: The system changes the edited item in the database and updates all lower level user views.

Stimulus: Administrator wants to delete an item.

Response: The system deletes the item in the database and updates all lower level user views.

3.4.3 Functional Requirements

These functions are very simple and mostly explained above:

- ToDo.add
- ToDo.edit
- ToDO.delete

4. Other Nonfunctional Requirements

4.1 Performance Requirements

Due to the system being web based, the performance of the system entirely depends on the speed of the internet connection. With a base speed of 25mbps, the system should do each request in < 1 second.

4.2 Safety Requirements

No safety requirements have been specified.

4.3 Security Requirements

- Every user must have a password to login to the software.
- Every user will have a user level that dictates what they can/can’t see in the views.
    - These permission levels will be set by the administrator.

4.4 Software Quality Attributes

Portability: It is important that the software can be used from outside the office. To ensure the product can be reached from anywhere with an internet connection, the software will be a web application to be used in browsers.

Reliability: In order to stay reliable, the software will be hosted on an AWS server.

Usability: The system will be as simple as possible. This is so any new employee will be able to understand how to use it.

Maintainability: Since the software will be hosted on an AWS server, it will be easy to maintain from anywhere in the world with an internet connection. Developers will not need to be on site to maintain the product.

5. Other Requirements

Appendix A: Glossary

Appendix B: Analysis Models

In a separate document.

Appendix C: To Be Determined List

None at this time.