

# **Software Requirements Specification**

**For**

# **Right-On-Time**

**Version 1.0**

**Prepared by Mikel Millard**

**November 10<sup>th</sup>, 20**

## TABLE OF CONTENTS

Table of Contents.....	ii
Revision History .....	iii
1. Introduction .....	1
1.1 Purpose .....	1
1.2 Intended Audience.....	1
1.3 Scope .....	1
1.4 Document Conventions .....	2
2. Overall Description.....	3
2.1 Product Perspective .....	3
2.2 Product Functions.....	3
2.3 User Characteristics .....	4
2.4 Operating Environment.....	4
2.5 Constraints .....	4
2.6 User Documentation .....	4
2.7 Assumptions and Dependencies .....	5
2.8 Appropriating Requirements .....	5
3. External Interface Requirements .....	6
3.1 User Interfaces .....	6
3.2 Hardware Interfaces .....	6
3.3 Software Interfaces.....	6
3.4 Communications Interfaces .....	7
4. System Features .....	8
4.1 User Class – Admin .....	8
4.1.1 Create Project .....	8
4.1.2 Edit Project.....	8
4.1.3 Finish Project .....	9
4.1.4 Create BOM .....	9
4.1.5 Edit BOM .....	10
4.1.6 Purchased BOM.....	10
4.1.7 Create Employee .....	10
4.1.7 Edit Employee.....	11
4.1.8 Delete Employee.....	11
4.1.8 Create To-Do list .....	12

4.1.9 Edit To-Do List .....	12
4.2 User Class – Employee .....	13
4.2.1 Add Hours .....	13
4.2.2 Edit Hours .....	13
4.2.3 Create BOM .....	13
4.2.4 Edit BOM .....	14
5. Other Nonfunctional Requirements .....	15
5.1 Performance Requirements .....	15
5.2 Safety Requirements .....	15
5.3 Security Requirements .....	15
5.4 Software Quality Attributes .....	15
5.4.1 Reliability .....	15
5.4.2 Maintainability .....	15
5.4.3 Portability .....	16
5.4.4 Usability .....	16
5.4.5 Scalability .....	16

## REVISION HISTORY

Name	Date	Reason For Changes	Version

## 1. INTRODUCTION

This section will help give a brief understanding and overview of the SRS document. There will also be a list of useful abbreviations and definitions that might be used throughout this document.

### 1.1 PURPOSE

The purpose of this SRS document is to give a thorough description of the requirements for “Right-On-Time” software. There will be descriptions of the purpose of this project as well as details on the whole development of the software. There will be information on user/admin interactions, user/admin interfaces, as well as system constraints and design choices.

### 1.2 INTENDED AUDIENCE

The client, Breedt Production Tooling and Design, is the main audience this document is prepared for. It is here to give the client a detailed idea of how the software will be developed. This document will be proposed to the client for their approval. It will also serve as a reference for developing the first version of the system for the development team.

### 1.3 SCOPE

“Right-On-Time” is a web application that allows the client to keep track of what kind of hours the employees are putting into projects, whether they be machine hours, R&D hours, etc. to insure the outcome of the project has a monetary net gain. This application will be used solely by the client, Breedt Production Tooling and Design.

The client will have an admin user level to have an overview of all the current projects being ran and the employees’ information on the hours being put into the project and the bill of materials for every project. There should also be detailed overviews of the projects such as the project name, customer, and deadline as well as information on the amount of money and hours that have been put into that project. A to-do list will also be incorporated for the admins to make at the end of the day as a simple list of the important things to get done the next day.

The employee user level will employees enter the hours they are putting into specific projects they are working on so that each different type of hour can be billed to the customer correctly and efficiently, as well as give the tasks each employee is currently working on. It will also give the employees an easy access to ordering materials and seeing if those materials have been ordered/delivered.

The overall goal of this application is to save the client money by making sure they aren’t coming to a loss with how they bill their customers and to make sure they aren’t wasting materials. It will also help keep track of the tasks their employees are working on more efficiently to make sure that more time than needed is being spent on certain aspects of each project.

## 1.4 DOCUMENT CONVENTIONS

The following list is meant to be a reference to the reader to better understand the document. This list will contain all abbreviations and acronyms that might be used throughout the document as well as any useful definitions to give the reader a better understanding of the document.

Term	Definition
User	Employees or Admins that will interact with the web application
Admin/Administrator	System administrator who has the permissions to manage and control the system
BPTDesign	The client, Breed Production Tooling and Design
Browser	Internet browsers that the system will be compatible with
DESC	Description
RAT	Rational
DEP	Dependency
BOM	Bill of materials
GE	General Engineering
GM	General Machine Shop
GF	General Fabrication Shop
GB	General Building
SP	Special Project
TD	Tooling Design – Client
TDB	Tooling Design – BREEDT
TR	Tooling Repair – Client
TRB	Tooling Repair – BREEDT
PD	Product Development – Client
PDB	Product Development – BREEDT
EDU	Education, Interns
AP	Apprentices

## 2. OVERALL DESCRIPTION

This section is meant to give a more technical overview of the system. The system will be explained in context to show how the system might interact with the employees and admins as well as introduce the basic functionality.

### 2.1 PRODUCT PERSPECTIVE

This web application will be a new, self-contained product meant specifically for BPTDesign. It will be made as a web application that is optimized for mobile browsing as well. It will be used as an organization tool to better keep track of the types of hours spent on each project and the bill of materials that are needed for each project.

The web application will communicate with a database that holds most of the information. This database should hold information about the specific projects that the company is currently working on, such as the project name, customer, deadline, project code. It should also hold employee information such as the hours spent on projects and what type of hours they are and their employee id. Communicating with the database, the web application will be able to access all the information needed to display the information both the administrators and the employees need to see during the projects. The admin level users will be able to add and modify information inside the database while the employee level users will just be able to get the data in the database.

Since we do want to optimize this web application for mobile use to benefit the employees more, there will be some restrictions on resource allocation. We will need to make sure that the processes of the application aren't slowed down when used on a mobile device to insure that the employees are being efficient in their work time and don't have to spend a lot of their time reporting their hours on certain tasks.

### 2.2 PRODUCT FUNCTIONS

The web application must let the admin level user create and modify projects and that projects information. Since the nature of BPTDesign is to have new and different projects all the time, this will be a very essential function for the admin level user. The admin should also be able to add certain employees to different projects, limiting their access to only that projects information and creating a BOM the projects they are on. Managing the BOM to make sure the materials get will also be needed, as well as the ability to create a to-do list for the next work day.

The employee user level will let the employee get some of the project information such as the name and the project number of the projects they are working on to document what kind of hours they are putting into certain projects. They will need to be able to also access the job codes and create a BOM for the projects they have been given access to by the admins. The employee also needs to be able to view all their active tasks to make sure they track the hours accurately.

The database will store all the project information that is added by the admin level user and viewed by the employee level user. The database should only be editable by the admin level user.

## 2.3 USER CHARACTERISTICS

There will be two types of users of this system: admin level users and employee level users. Both types have a different use of the system and slightly different objectives.

Admin level users are there to create/edit/delete projects and all the information associated with those projects. They will need to be able to view all the employees' hours on projects and what tasks they are doing on those projects. Creating/Editing/Ordering BOMs will also be conducted by this user, but the employee level user will also have some access to creating BOMs.

Employee level users will only be able to view the projects and the information associated with that project for the projects that they are given access to by the admins. They need to be able to track their hours spent on projects and what types of hours they spent on the project.

## 2.4 OPERATING ENVIRONMENT

The software will operate as a web application hosted on a server. It will be optimized to be primarily accessed on different mobile devices as well as different browsers. There will also be a database that the web application will need to be able to communicate with.

## 2.5 CONSTRAINTS

The web application will be constrained by the different mobile devices it will need to support. We must allocate resources with the major mobile devices in mind, making sure that the speed of the worst mobile device is still at in an acceptable place.

The web application will also have some internet connection restraints. Since the application will need to communicate with a database to access certain information, the users will need to have an internet connection. The speed of such connection could affect the speed of the whole application.

With the database in mind, the application will also be constrained by the capacity of the database. There will be many projects, as well as a ton of information about each project and each employee that all need to be stored in the database. The more data that the database holds, the longer it might take to fetch certain information that is requested slowing down the whole process as well.

## 2.6 USER DOCUMENTATION

For user documentation, we will provide a manual for both the admin level users and the employee level users to help them learn how to use the application and for a reference for any questions that may arise down the road. We will also provide a small tutorial session for the admin level users and then a separate one for the employee level users, since their interactions with the application will be different.

## 2.7 ASSUMPTIONS AND DEPENDENCIES

One major assumption is that the web application will be used mostly on mobile devices, and these mobile devices have enough performance to run the application. In some cases, where the device doesn't have enough resources available, the application might not work as intended, if at all.

Another assumption is that the users will always have access to an internet connection, since that will be necessary to communicate with the database. We must also assume that the internet connection can handle the load of information traveling between the database and the web application.

## 2.8 APPROPRIATING REQUIREMENTS

In the case that the project is delayed, we must shift around certain features to make sure the most important functions can make it in the first version. Any additional features that need to be added to improve functionality can be developed in later versions of the software.



### 3. EXTERNAL INTERFACE REQUIREMENTS

This section will provide more details on all the inputs into and outputs from the web application. It will also give descriptions of the hardware, software and communication interfaces.

#### 3.1 USER INTERFACES

The admin level user, upon opening the web application, should see a login screen to confirm their credentials and give them the correct permissions. Upon logging in, the admin should see a dashboard that has a list of the projects that are currently active, along with buttons to create new projects and to modify/finish current projects. The admin should also be able to expand on the projects to get more details about them, such as the customer and the employees that are working on those projects.

The admin should also see a section for the BOM that are currently in progress, as well as buttons to create new BOM and modify/finish current ones. Each BOM should have a BOM code that corresponds it to an open project so that there aren't BOM open for projects that are closed.

The employee level user should also see a login screen upon opening the web application. Once they have confirmed their credentials and have the correct permissions, they should see a list of projects that they have access to and are currently working on. Upon expanding these projects, they should be able to see the hours they have put in certain areas of the project as well as buttons to add new hours for other tasks on that project.

The employees should also see information on BOM that correspond only to the projects they are working on. They should be able to create and modify BOM and upon expanding on them they should be able to see the details on what is on the BOM.

#### 3.2 HARDWARE INTERFACES

Since web application doesn't designated hardware, it does not have any direct hardware interfaces. The hardware connection to the database server is managed by the browsers on the mobile phone and web server.

#### 3.3 SOFTWARE INTERFACES

The web application must work with multiple web browsers and on multiple mobile devices. However, since the software is a web application, the operating systems on the mobile devices won't be as important as the web browsers that those mobile devices will use to view the application.

### 3.4 COMMUNICATIONS INTERFACES

Even though communication between the database server and the web application is very important, there won't be any communication interfaces needed. The communication should be handled by the mobile devices web browsers and the web server.

## 4. SYSTEM FEATURES

This section goes over the fundamental actions of the software for all the users.

### 4.1 USER CLASS – ADMIN

This section will specify the actions that will be given to the admins of the software.

#### 4.1.1 CREATE PROJECT

##### *4.1.1.1 DESCRIPTION AND PRIORITY*

This feature allows the admin to create a new project with all the needed information. This feature is of a high priority.

##### *4.1.1.2 STIMULUS AND RESPONSE*

Stimulus: The admin wants to create a project.

Response: The admin is prompted by the system to input the necessary information for the project. The system will then create the project object with the given information to be stored in the database to be accessed later.

##### *4.1.1.3 FUNCTIONAL REQUIREMENTS*

Project.create: This will allow the admin to create the new project.

#### 4.1.2 EDIT PROJECT

##### *4.1.2.1 DESCRIPTION AND PRIORITY*

This feature will allow the admin to edit a current project. This feature of a high priority.

##### *4.1.2.2 STIMULUS AND RESPONSE*

Stimulus: The admin wants to edit a current project.

Response: The system changes the information in the database to the new information that the admin specified.

##### *4.1.2.3 FUNCTIONAL REQUIREMENTS*

Project.editID: Allows the admin to edit the project ID.

Project.add: Allows the admin to add employees to the project.

Project.remove: Allows the admin to remove employees from the project.

Project.editName: Allows the admin to edit the name of the project.

Project.editCust: Allows the admin to edit the customer for the project.

Project.editPO: Allows the admin to edit the PO #.

### 4.1.3 FINISH PROJECT

#### *4.1.3.1 DESCRIPTION AND PRIORITY*

This feature will allow the admin to mark a project as finished. This feature is of high priority.

#### *4.1.3.2 STIMULUS AND RESPONSE*

Stimulus: The admin wants to mark a project as finished.

Response: The admin is prompted by the system to specify which project they want to mark as finished.

#### *4.1.3.3 FUNCTION REQUIREMENTS*

Project.finish: Allows the admin to mark a project as finished.

### 4.1.4 CREATE BOM

#### *4.1.4.1 DESCRIPTION AND PRIORITY*

This feature allows the admin to create a BOM for a project. This features of high priority.

#### *4.1.4.2 STIMULUS AND RESPONSE*

Stimulus: The admin wants to create a new BOM for project.

Response: The admin is prompted by the system to enter the materials in the BOM and for which project the BOM is for.

#### *4.1.4.3 FUNCTIONAL REQUIREMENTS*

Bom.create: Allows the admin to create a BOM for a project.

#### 4.1.5 EDIT BOM

##### *4.1.5.1 DESCRIPTION AND PRIORITY*

This feature allows the admin to edit a BOM. This feature is of high priority.

##### *4.1.5.2 STIMULUS AND RESPONSE*

Stimulus: The admin wants to edit a BOM.

Response: The system changes the parts of the BOM to the new information that was specified by the admin.

##### *4.1.5.3 FUNCTIONAL REQUIREMENTS*

Bom.editMat: Allows the admin to edit the materials on the BOM.

Bom.editProject: Allows the admin to edit which project the BOM is associated with.

Bom.editNotes: Allows the admin to edit the notes on the BOM.

Bom.editPrice: Allows the admin to edit the price on the BOM.

#### 4.1.6 PURCHASED BOM

##### *4.1.6.1 DESCRIPTION AND PRIORITY*

This feature allows the admin to set the BOM to purchased. This feature is of high priority.

##### *4.1.6.2 STIMULUS AND RESPONSE*

Stimulus: The admin wants to set the BOM to purchased.

Response: The system prompts the admin to select which BOM to set as purchased.

##### *4.1.6.3 FUNCTIONAL REQUIREMENTS*

BOM.purchased: Allows the admin to set a BOM to purchased.

#### 4.1.7 CREATE EMPLOYEE

##### *4.1.7.1 DESCRIPTION AND PRIORITY*

This feature allows the admin to create a new employee in the database. This feature is of medium-high priority.

#### *4.1.7.2 STIMULUS AND RESPONSE*

Stimulus: The admin wants to create a new employee into the database.

Response: The system prompts the admin to enter the required information for the employee then adds that new employee to the database.

#### *4.1.7.3 FUNCTIONAL REQUIREMENTS*

Employee.create: Allows the admin to create a new employee.

### **4.1.7 EDIT EMPLOYEE**

#### *4.1.7.1 DESCRIPTION AND PRIORITY*

This feature allows the admin to edit the information of an employee in the database. This feature is of medium priority.

#### *4.1.7.2 STIMULUS AND RESPONSE*

Stimulus: The admin wants to edit information of an employee.

Response: The system changes the employee's information to the new information the admin entered.

#### *4.1.7.3 FUNCTIONAL REQUIREMENTS*

Employee.editPerm: Allows the admin to edit the projects the employee has permissions for.

Employee.editID: Allows the admin to change the employees ID.

Employee.editName: Allows the admin to edit the name of the employee.

### **4.1.8 DELETE EMPLOYEE**

#### *4.1.8.1 DESCRIPTION AND PRIORITY*

This feature allows the admin to delete employees from the database. This feature is of medium priority.

#### *4.1.8.2 STIMULUS AND RESPONSE*

Stimulus: The admin wants to delete an employee from the database.

Response: The system prompts the admin with which employee to delete and deletes that employee from the database.

#### *4.1.8.3 FUNCTIONAL REQUIREMENTS*

Employee.delete: Allows the admin to delete the specified employee.

### **4.1.8 CREATE TO-DO LIST**

#### *4.1.8.1 DESCRIPTION AND PRIORITY*

This feature allows the admin to create a new to-do list for the next day. This feature is of medium-low priority.

#### *4.1.8.2 STIMULUS AND RESPONSE*

Stimulus: The admin wants to create a new to-do list for the next day.

Response: The system adds a new to-do list to the database with the tasks specified from the admin.

#### *4.1.8.3 FUNCTIONAL REQUIREMENTS*

ToDoList.create: Creates a new to do list with specific tasks

### **4.1.9 EDIT TO-DO LIST**

#### *4.1.9.1 DESCRIPTION AND PRIORITY*

This feature allows the admin to edit a to-do list. This feature is of low priority.

#### *4.1.9.2 STIMULUS AND RESPONSE*

Stimulus: The admin wants to edit a to-do list.

Response: The system edits the tasks on the specified to-do list.

#### *4.1.9.3 FUNCTIONAL REQUIREMENTS*

ToDoList.editTasks: Allows the admin to edit the tasks on a to-do list.

## 4.2 USER CLASS – EMPLOYEE

This section will specify the actions available to the employees using the software.

### 4.2.1 ADD HOURS

#### *4.2.1.1 DESCRIPTION AND PRIORITY*

Allows the employee to add hours to a project. This feature is of high priority.

#### *4.2.1.2 STIMULUS AND RESPONSE*

Stimulus: The employee wants to add hours of a specific type.

Response: The system creates an entry into the database storing the specific hours the employee added for a specific project.

#### *4.2.1.3 FUNCTIONAL REQUIREMENTS*

Hours.add: Allows the employee to add hours of a specific type to a specific project.

### 4.2.2 EDIT HOURS

#### *4.2.2.1 DESCRIPTION AND PRIORITY*

This feature allows the employee to edit specific hours on specific projects in case an error was made when originally reporting them. This feature is of medium priority.

#### *4.2.2.2 STIMULUS AND RESPONSE*

Stimulus: The employee wants to edit specific hours on a specific project.

Response: The system makes the appropriate changes in the database entry for those hours.

#### *4.2.2.3 FUNCTIONAL REQUIREMENTS*

Hours.editType: Allows the employee to edit the type of hours reported.

Hours.editAmount: Allows the employee to edit the number of hours reported.

Hours.editProject: Allows the employee to edit the project the hours were reported on.

### 4.2.3 CREATE BOM



#### *4.2.3.1 DESCRIPTION AND PRIORITY*

This feature allows the employee to create a BOM for projects they have permissions on. This feature is of high priority.

#### *4.2.3.2 STIMULUS AND RESPONSE*

Stimulus: The employee wants to create a BOM for a project they are on.

Response: The system creates a database entry holding the necessary information for the BOM the employee wanted to create.

#### *4.2.3.3 FUNCTIONAL REQUIREMENTS*

Bom.create: Allows the employee to create a BOM for a project they are on.

### **4.2.4 EDIT BOM**

#### *4.2.4.1 DESCRIPTION AND PRIORITY*

This feature allows the employee to edit a BOM that they have created. This feature is of medium-high priority.

#### *4.2.4.2 STIMULUS AND RESPONSE*

Stimulus: The employee wants to edit a BOM that they have created.

Response: The system makes the specified changes to a BOM that they employee has created.

#### *4.2.4.3 FUNCTIONAL REQUIREMENTS*

Bom.editMats: Allows the employee to edit the materials on the BOM.

Bom.editProject: Allows the employee to edit what project the BOM is for.

Bom.editNotes: Allows the employee to edit the notes on the BOM.

## 5. OTHER NONFUNCTIONAL REQUIREMENTS

This section is to provide more details on the nonfunctional requirements such as performance, safety, etc.

### 5.1 PERFORMANCE REQUIREMENTS

For performance, we want the web application to run as fast and smooth as it possibly can. There are limitations in what mobile devices can do in terms of performance, but we want to keep this in an acceptable range of less than 2 seconds for large data requests. We want the actions being performed to look as seamless as possible.

### 5.2 SAFETY REQUIREMENTS

There will need to be some safeguards in place to make sure that data does not get lost due to a server error or a communication error between the web application and the database. We also will want safeguards to help prevent accidental attempts to delete the wrong BOM, employee, etc.

### 5.3 SECURITY REQUIREMENTS

There will need to be safeguards to make sure the information gets transferred between the database and web application securely. There should also be a login process for anyone attempting to use the web application so that employees only get certain permissions and anyone who is not an admin or employee will not be able to access any of the information in the database.

### 5.4 SOFTWARE QUALITY ATTRIBUTES

#### 5.4.1 RELIABILITY

The web application needs to be reliable and work most the time. We want to minimize errors and crashes so that there is no loss of data or loss of work.

#### 5.4.2 MAINTAINABILITY

The web application should be easily maintainable, with some added features after the original version is released. We will want all maintenance to be done quickly so that there are no major down times of the application.

#### 5.4.3 PORTABILITY

The web application should be optimized to work on mobile devices as that is what most of the employees will be using when trying to access the web application. We want to accommodate as many different mobile devices as possible, as well as many browsers as possible.

#### 5.4.4 USABILITY

The web application needs to be easily usable by the admins and the employees. Everything should be clear and concise, with decently sized buttons to make pushing them easier for all users.

#### 5.4.5 SCALABILITY

The web application should scale well with most mobile devices. Since there are a lot of different dimensions for mobile devices, the web application needs to fit nicely to a wide majority of dimensions so that all users have a good experience with the web application.