# 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 document is to give an outline for the web application “Right-On-Time.” The document will go in depth about what the project is and how the user will interact with the application. Moreover, this document will lay out the requirements as well as assessing the validity, consistency, and prioritization of the requirements. The primary reason for this document is to give the users a schematic to validate the requirements and view models while giving the developers a guide during the development process.

## 1.2 Product Scope

In short, “Right-On-Time” is a project tracking system designed to keep Breedt Production Tooling and Design on track with both time and budget for their projects. The goal is to be able to determine if a project was actually profitable by tracking how many hours and how much money is being put into it. Another objective for “Right-On-Time” is to track how many billable hours each employee is actually completing. Finally, Breedt wants to formalize its process for purchasing project materials by requiring all purchases to be tracked through “Right-On-Time.” Overall, Breedt wants a better understanding of their project flow and expenses.

## 1.3 Definitions

**BPT Design** –BreedtProductionTooling and Design

**“Right-On-Time”** –Title of the product

**BOM** –BillofMaterials

**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 Intended Audience and Reading Suggestions

This document is designed to be a contract between Breedt Production Tooling and Design, and the developers of “Right-On-Time.” Therefore, the main intended audience for this document includes the developers of Right-on-Time and the owners of Breedt. The contents of this document will contain an overall description and scope of “Right-On-Time:, an overview of hardware and software interfaces, and finally a comprehensive list of required features. To fully grasp this software, I would recommend reading the entire document in order.

## 1.6 Team

Brandt DenHerder – Team Lead

Laura Leist – Database Lead

Mikel Millard – Documentation Lead

Robert Roppo – UI Lead

Charles Saeyang – Programming Lead

# 2.0 Overall Description

This section will describe the mechanics of “Right-On-Time”. It will show a description in a developer’s point of view via diagram and explanation, product functions on a user level, an in-depth description of different user levels, constraints, and assumptions.

## 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 will hold information about the specific projects that the company is currently working on, such as the project name, customer, deadline, project code. It will 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 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 are not slowed down when used on a mobile device to ensure that the employees are being efficient in their work time and do not have to spend a lot of their time reporting their hours on certain tasks.

## 2.2 Product Functions

Major High-level Functions

1. Project Tracking
   1. Tracking hours put towards each project
   2. Tracking bill of materials (BOM) items
2. Project Overview
   1. Visualize statistics on the amount hours dedicated to the project
   2. Project details: Name, customer, deadline, hours invested, money invested, other.
3. To-Do List
   1. Managers can create to-do list for the next day
   2. Employees can view to-do list on dashboard
4. Multiple User Levels

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Create Project | Enter BOM Item | Authenticate BOM Item | Add/Modify  Tasks | Modify To-Do list | View Budget Information | Modify Users |
| Admin | x | x | x | x | x | x | x |
| Shop Manager | x | x |  | x | x |  |  |
| Project Manager |  | x |  | x |  |  |  |
| Purchasing |  |  | x |  |  |  |  |
| Employee |  |  |  | x |  |  |  |

Table: This table shows the different user levels and what functions that user level can do.

## 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 Operating Environment

Right-on-Time is being developed as a web based application, which means the user interaction and interface can be conducted on any modern web browser connected to the internet. This includes but is not limited:

1. Internet Explorer
2. Chrome
3. Safari
4. Firefox

The database and backend code will be hosted through a service like Amazon Web Services, discussed in section 3.2 Hardware Interfaces.

## 2.5 Design and Implementation Constraints

Since we are developing “Right-On-Time” as a web application there are a few design constraints that come along with web development. These constraints are as follows:

1. The application must work on different browsers
   1. Since Right-on-Time must work on different browsers, we are constrained to UI functionality that works cross browser.
   2. Working with different browsers also requires testing the software on every browser we want to be able to access “Right-On-Time”
2. The application must have responsive design for mobile use
   1. Since browsers can be viewed as different sizes such as desktop, tablet, and mobile, our interfaces need to be designed such that the functionality will still fit no matter the screen size.
   2. This may also require entirely different interfaces for mobile vs. desktop.
3. We are also limited to the functionalities that browsers allow us. For instance, some browsers now support desktop notifications, while others do not.

## 2.6 Assumptions and Dependencies

For this SRS and throughout the development process, we will assume that all employees will have the technology and devices necessary 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.

## 2.7 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.

# 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

Right-on-Time will consist of six main views, and in this section we will list those views as well as what functionality will be available in each view. Each view will also contain a menu bar with quick access to the Dashboard View, the Projects List View, and the Settings View.

1. Login View
   1. The user will be able to enter a username and password to login to Right-on-Time
2. Dashboard View
   1. The user will be able to view their currently active tasks
   2. The user will be able to view the current to-do list
   3. The user will be able to view their recently completed tasks
   4. The user will be able to start/pause/complete their tasks
   5. If the user has a high enough user level they will be able to see a list of all active tasks
   6. If the user has a high enough user level, they will be able to edit the to-do list.
3. Projects List View
   1. The user will be able to see a list of all current projects with the newest at the top
   2. The user will be able to click on a project to see the project in the Project View
   3. If the user has a high enough user level, they will be able to add new projects
4. Project View
   1. The user will be able to see the project details include the following:
      1. Project Name
      2. Project ID
      3. Project Manager
      4. Project Description
      5. Project Status (Ready for work, or not ready for work)
      6. Current Hours Invested vs. Hours Budgeted
      7. Money invested (If the user level is high enough)
   2. The user will be able to see the projects currently active tasks
   3. The user will be able to see the projects current BOM list
   4. The user will be able to see the currently complete tasks for the project
   5. The user will be able to start/pause/complete their tasks for this project
   6. If the user has a high enough user level, they will be able to request BOM items
   7. If the user has a purchaser user level, they will be able to modify the status of BOM items
   8. If the user has a high enough user level, they will be able to modify the project details
5. Settings View
   1. The user will be able to see their current user information
   2. The user will be able to change their password
   3. If the user is an Administrator, they will be able to see a list of all employees and their user levels.
   4. If the user is an Administrator, they will be able to modify the user levels of employees.
   5. If the user is an Administrator, they will be able to add and remove employees
6. Employee View
   1. If the user has a high enough user level, they will be able to see all the information about a user. This includes the following:
      1. Employee’s active and recently completed tasks
      2. Employee’s name, username, and user level

## 3.2 Hardware Interfaces

The hardware interfaces will be handled by the hosting service we choose for Right-on-Time. The hosts we have discussed may be one of the following options:

1. Amazon Web Services
2. Microsoft Azure

Either option will have similar functionality when it comes to deploying and managing Right-on-Time.

## 3.3 Software Interfaces

Since Right-on-Time is being developed as a standalone web application, there will be no software interfaces developed for it.

## 3.4 Communications Interfaces

Since Right-on-Time is a web based application, all communication between the user and the application will be conducted through a web browser. All of the communication between the front-end application and the backend software will be done through HTTP requests. The application will handle all of this communication without user involvement. Also, since we are not dealing with payments, the HTTP communication standard should be secure enough.

# 4. System Features

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

## System Display Features

### View Project Details

|  |  |
| --- | --- |
| Description | This feature will allow the user to view the high level details about the project they are viewing. These details will be viewed on the specific project’s page.  These details include the following:   1. Project Name 2. Customer 3. Project ID/PO 4. Project Manager 5. Current Hours Invested 6. Budgeted Hours 7. Project Description 8. Current Budget Invested (Depending on User Level) 9. Projects Status (Either not ready to be worked on, or ready for work) |
| Validity | The employees need to be able to see information about the project they are adding tasks to, as well as who they need to talk to about the project if they have questions. These details will also show the employees if they are on track and within the budgeted hours. |
| Consistency | This requirement doesn’t conflict with any other specified requirement. |
| Feasibility | This will be possible by storing information in the database for each project and retrieving it when a user goes to the project page. |
| Error Handling | The only error possible would be if the project does not exist in the database. If that is the case, then we will send the user back to the projects list view with an error about missing project. |
| Priority | Required |

### View Active and Completed Tasks

|  |  |
| --- | --- |
| Description | This feature will allow users to view their tasks as well as all open tasks. Tasks will be shown on the users dashboard view, the project view, and individual employee view. Tasks can be shown in the following lists:   1. Employee’s Currently Active Tasks 2. Employee’s Completed Tasks 3. All Active Tasks 4. Project’s Active Tacks 5. Project’s Completed Tasks |
| Validity | Not only do employees need to be able to see what tasks they are currently working on, but they also need to be able to see what tasks they have completed for the day to judge how productive they have been. It is also important to be able to see what tasks are being worked on for each project, and for project managers and admins to be able to see how much work is being tracked each day. |
| Consistency | This requirement doesn’t conflict with any other specified requirement. |
| Feasibility | This will be possible by storing the tasks in the database for each project and applying queries to receive the proper lists of tasks. |
| Error Handling | There should not be any errors here as only tasks that are found in the database with the correct conditions for each list will be shown. |
| Priority | Required |

### View Bill of Material Items

|  |  |
| --- | --- |
| Description | This feature will show all of the current bill of material items that have been requested and purchased for a specific project. There are a few ways to filter this list by the following conditions:   1. Requested BOM items 2. Ordered BOM Items 3. Delivered BOM Items |
| Validity | One of the requirements to start working on a project is for the materials of the project to be available. It is easier to start a task if you know the material is ready to be worked on. Also, one of the major things to be tracked in this system in the amount of money being invested in each project. Tracking the cost of purchased materials will allow that to be possible. |
| Consistency | This requirement doesn’t conflict with any other specified requirement. |
| Feasibility | This will be possible by storing the BOM items in the database for each project and applying queries to receive the proper lists BOM items. |
| Error Handling | There should not be any errors here, as only BOM items which are in the database for the specific project will be shown. |
| Priority | Required |

### View To-do List

|  |  |
| --- | --- |
| Description | This feature will allow employees to see what the shops main goals are. This could include projects the shop is focusing on or tasks that are imperative to be completed soon. |
| Validity | This feature allows for BPTD to easily notify their employees about to the current priorities of the shop. |
| Consistency | This requirement doesn’t conflict with any other specified requirement. |
| Feasibility | This can be accomplished by keeping a list of to-do items in the database and retrieving them when a user access the dashboard. |
| Error Handling | There should not be any errors here, as only to-do items which are in the database will be shown. If there are no to-do list items in the database, then the to-do list will not be visible. |
| Priority | Ideal |

## System Functional Features

### Start/Pause/Finish Task Items

|  |  |
| --- | --- |
| Description | This feature will allow employees to add and modify their tasks. There will be a few different functionality’s surrounding adding and modifying tasks. Those functions are as follows:   1. Start Task 2. Pause Task 3. Complete Task 4. Delete Task |
| Stimulus Sequence | Each possible function surrounding tasks will have its own user input requirements as follows:   * Start Task: User must input the project code if they are not adding the task from the project view. The user must also input a description of the task. * Pause Task: The user must input a reason for pausing the task * Complete Task: The user just has to click complete on the task * Delete Task: The user will just have to click delete on the task |
| Validity | This feature relates to the main reason the application was proposed. Assuming employees enter tasks properly, this will allow BPTD to easily track the productivity of their employees as well as see how many hours are being invested in each project. |
| Consistency | This feature does not conflict with any other features, as it is one of the pain purposes for this application. |
| Feasibility | These features can be implemented by storing the tasks with their corresponding project in the database. The user level as well as the creator of the task will determine if the task can be edited by a user. |
| Error Handling | We will force the user to input all of the required fields when using any of the functions above. This will hopefully prevent any errors that could occur. |
| Priority | Required |

### Add/Remove To-do List Items

|  |  |
| --- | --- |
| Description | This feature will allow a user with the correct user level to add and remove items from the to-do list, which will be viewable on every user’s dashboard. The two functions these users will have access to include:   1. Adding To-do List Items 2. Removing To-do List Items |
| Stimulus Sequence | Each of the feature functions above will have its own user input requirements as follows:   * Add to-do list item: User must input a description of the to-do list item as well as a priority for that goal. * Remove to-do list item: User will be able to click a delete button on the list item to remove it. |
| Validity | To have a to-do list available, there needs to be a way to add and remove items from the list as the priorities of the shop changes over time. |
| Consistency | This requirement doesn’t conflict with any other specified requirement. |
| Feasibility | This will be possible by modifying the list of items in the to-do list which is stored in the database. |
| Error Handling | There should be no chance for an error as we will require the user to enter all of the specified inputs to continue. |
| Priority | Ideal |

### Create/Modify Project Details

|  |  |
| --- | --- |
| Description | This feature will allow employees of the correct user level to create projects for BPTD. The two possible functions these users will be able to take are as follows:   1. Add a Project 2. Modify the Project Details |
| Stimulus Sequence | Each of the feature functions above will have its own user input requirements as follows:   * When adding a project, the user will have to input the following details:   + Project Name   + Customer   + Project Type   + Project Manager   + Budgeted hours for the project   + Project Description * When modifying the project details, the user will be able to edit any of the details above, besides project type, and will also be able to edit the following additional detail:   + Status (Whether the project can or cannot be worked on yet. |
| Validity | Since one of the reasons Right-on-Time is being developed is to track how much time employees are putting into a project, there needs to be a way to track projects. This will allow the owners of BPTD to understand better what projects are being worked on. |
| Consistency | This feature does not conflict with any other features, as it is one of the pain purposes for this application. |
| Feasibility | Each project will get its own entry in the database with the project details above, as well as lists of tasks and BOM items that are assigned to the project. Since this feature is one of the core pillars of Right-on-Time, many other features will be based on this implementation. |
| Error Handling | When adding projects, the information above will be required before submission, which should prevent any possible errors. |
| Priority | Required |

### Request/Modify Bill of Material Items

|  |  |
| --- | --- |
| Description | This functionality will allow users, of the correct user level, to request bill of material items for specific projects. The possible functions are as follows:   1. Request BOM items 2. Modify the status of BOM items |
| Stimulus Sequence | Each of the feature functions above will have its own user input requirements as follows:   * When requesting a BOM item, the user must have a user level higher than project manager and must enter a description and estimated cost of the requested item. * When modifying the BOM status, the user must have a user level of admin or purchasing and must choose a status from the following list of options:   + Requested   + Ordered   + Delivered   The user may also change the cost of the BOM item if the actual cost was different from the estimated cost. |
| Validity | Since one of Right-on-Time’s goals was to track the amount of money going into projects, this features is rather imperative. |
| Consistency | This feature does not conflict with any other features, as it is one of the pain purposes for this application. |
| Feasibility | The BOM list will be stored in the database under the project which it belongs to. In other words, each project will have a list of BOM items associated to it. |
| Error Handling | When adding or modifying the status of BOM items, the information above will be required before submission, which should prevent any possible errors. |
| Priority | Required |

### Add/Remove Employee

|  |  |
| --- | --- |
| Description | This feature will allow a user, of the user level admin, to add and remove users from the database of company employees. The two functions of this feature include the following:   1. Add user to the Right-on-Time user database 2. Remove user from the Right-on-Time user database |
| Stimulus Sequence | Each of the feature functions above will haves its own user input requirements as follows:   * When adding a user, the following information will be required:   + Name   + Username   + User level   + Temporary password * When removing a user, the admin will have to authenticate with their password. |
| Validity | It is very imperative of the application to track different users so that quality information about BPTD can be monitored. For this to be possible, it must also be possible for users to be added and removed from Right-on-Time. It is also important to restrict users to functions specified by BPTD, and user accounts with user levels will allow for that functionality. |
| Consistency | This feature does not conflict with any other features. |
| Feasibility | This feature will be accomplished by having a list of users and their information in the database. |
| Error Handling | When adding or removing users, the above inputs will be required which should reduce the chance of anything going wrong. |
| Priority | Required |

### Change Employee User Level

|  |  |
| --- | --- |
| Description | This feature will allow a user, of the user level admin, to modify another employee’s user level. The only function involved with this feature is as follows:   1. Change an Employee’s User Level |
| Stimulus Sequence | The functionality above will have the following input requirements:   * When changing an employee’s user level, the admin must choose from the following list of user levels:   + Administrator   + Shop manager   + Project manager   + Purchasing   + Employee |
| Validity | To restrict what users will be able to do within Right-on-Time, there needs to be a way to track what their access level is. This method of assigning a user level to each employee will give BPTD that functionality. |
| Consistency | This feature does not conflict with any other features. |
| Feasibility | This will simply be done by changing the user level field in the database for the user in question. |
| Error Handling | When modify an employee’s user level, the administrator will be required to choose from the list of user levels defined above. This should prevent any errors. |
| Priority | Required |

### Change Password

|  |  |
| --- | --- |
| Description | This feature will allow users to change their password. The only function involved with this feature is as follows:   1. Change a user’s password |
| Stimulus Sequence | The functionality above will have the following input requirements.   * The user must enter their new password two times |
| Validity | This is a function that is almost always required when dealing with user logins. It is imperative for the users to be able to change their password for security reasons alone. Also when a new employee is created, the admin knows their password, so the new employee needs a way to change it. |
| Consistency | This feature does not conflict with any other features. |
| Feasibility | This will be done by first checking to make sure the two new password inputs are the same, then modify the password field for the user in the database. |
| Error Handling | The only error is if the two password inputs are not the same, then the application will warn the user and let them try again. |
| Priority | Required |

# 5. Other Nonfunctional Requirements

This section is to provide more details on the nonfunctional requirements such as performance, safety, security and software quality attributes.

## 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 operations as quick as possible with no noticeable lag.

## 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 System Attributes**

**5.4.1 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.

**5.4.2 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.

**5.4.3 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.

**5.4.4 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.

9

**5.4.5 Portability**

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