

# Thrifty

## Software Requirements Specification

Edition 1 — October 2020

by Emin Okic, Logan Barrett, & Matt Rayman  
Iowa State University

These are the software requirements for the Thrifty mobile apparel merchandising application.  
This document will provide sections for each of the requirements needed.

### ***Requirements Types***

*Functional Requirements* are the fundamental or essential subject matter of the product. They describe what the product has to do or what processing actions it is to take.

*Nonfunctional Requirements* are the properties that the functions must have, such as performance and usability. Do not be deterred by the unfortunate type name (we use it because it is the most common way of referring to these types of requirements)—these requirements are as important as the functional requirements for the product’s success.

*Project Constraints* are restrictions on the product due to the budget or the time available to build the product.

*Design Constraints* impose restrictions on how the product must be designed. For example, it might have to be implemented in the hand-held device being given to major customers, or it might have to use the existing servers and desktop computers, or any other hardware, software, or business practice.

*Project Drivers* are the business-related forces. For example, the purpose of the project is a project driver, as are all of the stakeholders—each for different reasons.

*Project Issues* define the conditions under which the project will be done. Our reason for including them as part of the requirements is to present a coherent

---

### **1. The Purpose of the Project**

---

#### **1a. The User Business or Background of the Project Effort**

The Thrifty mobile apparel merchandising software will have the ability for users to buy, sell, and trade clothes all within the reach of their fingers.

The Thrifty mobile apparel merchandising software is aimed at being more environmentally friendly and customizable than other such products currently on the market. Thrifty will be a cross platform software available on one main application. Users can access the application from multiple devices at once.

#### **1b. The Goals of the Project**

Upon opening the application, users will be designated to a login or sign up screen depending on what the user requires. This will be determined based on prior logins from the current IP address. While logging in, the user will only be required to enter a username and password. If the fields are not filled in correctly, or are empty, Thrifty will notify the user of such warnings.

If the user happens to be a first-time user, then the user will then be led to a sign-up screen where they fill in the required information. Upon signing up, the user will be led to an introductory screen describing the app and its features. After completion, the user will then be able to begin using their account where they can post or browse items on the app.

## **2. The Client, Customer, and Other Stakeholders**

---

### **2a. The Client**

The Thrifty apparel merchandising system software will allow fast fashion influenced companies to resell, and better market non-trending clothes. In addition, individual users will be able to perform similar actions to make it easier for users to thrifty effectively.

The EverGreen system is aimed at being more water-efficient and customizable than other such products currently on the market. Evergreen has two modes: manual and automated. Homeowners can switch between these two via a control panel.

### **2b. The Customer**

Our customer is fashion enthusiasts looking to expand their wardrobe and resell unused clothing items. We are looking to solve problem of wasted clothing inside the fast fashion market.

This is because many people today and, in the future, will be require clothes. Therefore, we can safely say that not everyone will want to shop and buy new clothes every time. However, many will look to have someone do it for them or an efficient system to do it themselves.

### **2c. The Other Stakeholders**

Other stakeholders could be fast-fashion businesses and companies such as Zara looking to decrease losses on unused or non-trending clothing currently in stock. Clothing may be more or less expensive depending on location.

An example of this would be companies in the fast-fashion industry looking to lower their overall losses and waste but may not have the appropriate solution to do so.

### 3. Mandated Constraints

---

#### 3a. Solution Constraints

This problem must be solved using pre-existing software, and a device to run the application. The reason for this is because many apparel buyers will look for a central location to buy, sell, and reuse clothes provided by the Thrifty apparel merchandising software solution.

Thrifty provides two options for clothing items: upload and buy items. Users can switch between these two via selecting different options through the home menu.

While browsing items, the user sets parameters via the software to program what items appear on the current list and what items are to be prioritized later. Users can enter specific category parameters to schedule what items appear on the current browse. When the user selects an item, it will load a live message platform for the buyer and seller to communicate on.

When the user uploads an item, it will appear on their feed and the browsing list. When the area of location is too great, Thrifty will divide each location to their own appropriate locations for the user to select from. The home menu will display options such as the user profile, browsing screen, and an upload item icon for the user to select from. These options will help reflect real-time changes and increase user-friendliness.

The Thrifty apparel merchandising software solution shall operate using a device able to load the application. These devices currently include; IOS, Android, and Web Deployment opportunities. The client wishes to use a single source screen to access all the application features. All commands selected by the client shall respond with its appropriate result.

We want to consider software that imposes as little constraints as possible. Therefore, adding the need for a cross-platform application and build system is important because the client and users will not notice the constraints as much as the developer working directly with the software.

### **3b. Implementation Environment of the Current System**

The Thrifty apparel merchandising software solution shall operate functions and interfaces in an appropriate hardware-device environment.

The client wishes to use a single source panel to access all the applications features located in the software solution. However, the client will need the software to design and implement user set parameters and constraints to communicate with interfaces in order to have the system properly function. This in return, defines our environmental constraints for the Thrifty software solution implementation.

## **4. Naming Conventions & Terminology**

---

### **4a. Definitions & Data Dictionary**

---

Thriftling: A form of apparel merchandising where used clothes are sold or bought.

Cross-Platform: The software is a user-friendly device composed of multiple components that allow the user to easily control the given software through the appropriate device.

Display status: The home screen shows the current status of the Thrifty apparel merchandising software solution and related.

Parameters: These are the variables, or the arguments used to return correct results on the control panel.

## **5. Relevant Facts and Assumptions**

---

### **5a. Facts**

---

Apparel merchandising software solutions can help to increase revenue up to 11% of as compared to manually thrifting and buying clothes.

The average American citizen tends to throw away approximately 80lbs of clothing each year, contributing to 5% of the worlds landfill space. This leads to an overall of 12.8 million tons of textiles lost annually.

Fast-fashion corporations report up to 107 billion USD in losses and returns every year.

## **5b. Assumptions**

---

With global warming becoming an increasing issue, it is more important now than ever.

We must assume our software has the ability to perform on multiple devices allowing the user to access the application from nearly anywhere.

The Apparel merchandising software solutions must provide clothing set by the users buying decisions.

The Apparel merchandising software solutions must provide a list of users profiles to select from upon being searched.

## Functional Requirements

---

## **6. Scope of the Work**

---

### **6a. The Context of Work**

---

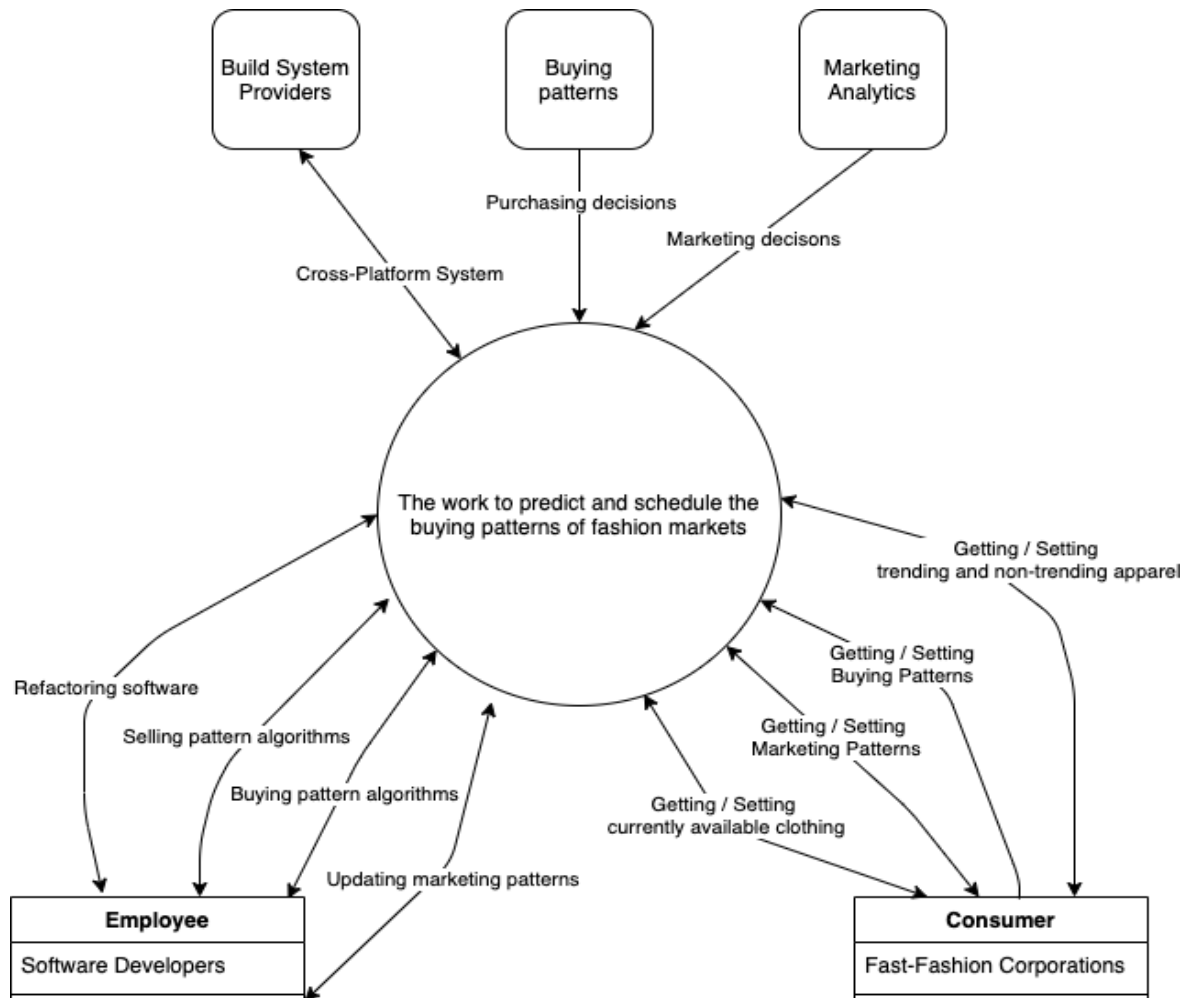
The Thrifty apparel merchandising software solution must provide clothing set by the users buying decisions.

The Thrifty apparel merchandising software solution is aimed to help reduce individual waste produced by the average user.

The Thrifty apparel merchandising software solution is aimed to benefit fast-fashion corporations in providing an easily accessible 3<sup>rd</sup> party software for the company to provide unwanted or non-trending clothes on.

## 6b. The Context of the Work Model

---



## 7. Scope of the Software Product

---

### 7a. Product Use Case List

This product use case diagram defines the boundaries for the scope above. It is commonly used in requirements engineering to better understand the study of work.

The Thrifty use-case diagram is supplemental to further explaining and quantifying the amount of work that goes into predicting, and scheduling watering times.

**Product use case name:** Set parameters

**Trigger:** User changes parameter on the categories screen

**Actor:** Thrifty user

**Steps:**

1. Updates the value of the buying pattern in the memory
2. Finds where the parameters are used
3. Uses the new value of the parameter
4. Tells user the parameter was updated

**Product use case name:** User Feed Status

**Trigger:** User follows another user

**Actor:** Thrifty user

**Steps:**

1. Gets the statuses of the current user and searched up user matching the key given
2. Gets the related parameters from the environment
3. Displays the statuses and updates on the current user's profile screen

**Product use case name:** Updating Marketing Analytics

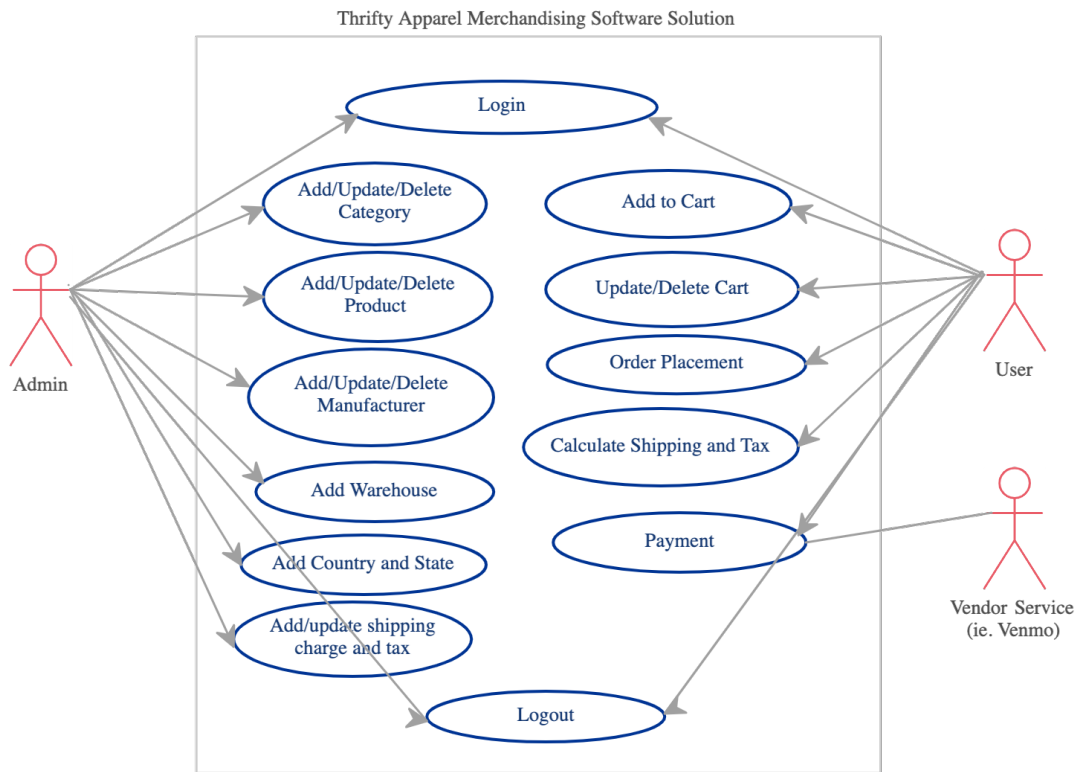
**Trigger:** User selects multiple items over time

**Actor:** Thrifty user

**Steps:**

1. Gets the statuses and tags of the selected items
2. Updates the state of the tags prioritized in a zone in the memory
3. Identify the buying patterns in the specified zone
4. Sort clothing appropriately in the specified zone
5. Tells user the tags of the clothing in the specified zone

## 7b. Product Use Case Model



Use Case

## 8. Functional Requirements

---

### 8a. Functional Requirements

---

**“The apparel merchandising software solution shall get the buying patterns, selling patterns, and current location of the apparel merchandising software solution.”**

*Rationale:*

We intend to make the customer’s lives easier not harder. Therefore, we need to define a limit with how customizable the marketing and analytics options are to avoid overwhelming the consumer.

*Fit Criterion:*

Too Many Features Hurts Usability, and Scalability. Creating software performs a small number of features exceedingly is a better experience for the user.

In addition, it makes the customer more likely to adopt updated versions of the software in the future.

---

**“When commanded by the user, the apparel merchandising software solution shall automatically override current location, and or current categories.”**

*Rationale:*

If the user desires to manually select specific clothing categories, we shall make it readily available for them to override the current marketing algorithm in order to display clothes they may desire more.

*Fit Criterion:*

Users shall be able to override the currently available items list to modify or replace a buying pattern or decision immediately upon using the categories feature/screen.

---

**“When commanded by the manufacture, the apparel merchandising software solution shall automatically reschedule available items and remove appropriate clothing items as needed.”**

*Rationale:*

It is within reason that there will be users not so interested in certain clothing styles. Therefore, we implement an automatic items available schedule based on the manufacture’s available clothing, trending clothing, and clothing past it’s expected purchase date.

*Fit Criterion:*

Users shall have the option to automatically schedule items available to them. These schedules shall have a system in place to adjust the schedule in the event the current schedule is overridden by the user when entering the category settings.

---

## 8b. Data Requirements

Figure 1 is a context diagram describing the model that effectively displays the work of the Thrifty apparel merchandising software solution.

Figure 2 is a use-case diagram describing the model that effectively shows the specific work needed for the Thrifty apparel merchandising software solution.

## Nonfunctional Requirements

---

## 9. Look and Feel

---

### 9a. Ease of Use Requirements

**“The software must display statuses clearly.”**

The Thrifty apparel merchandising software solution should have a home screen that clearly displays all the information. This will improve the client satisfaction and limit the number of mistakes.

## 10. Usability and Humanity

---

### 10a. Ease of Use Requirements

**“The software solution shall be modular for easy updates.”**

The Thrifty apparel merchandising software solution must be prepared to readily deliver and exchange software interfaces in order to maintain client satisfaction.

**“The parameter values shall be easy to update.”**

The Thrifty apparel merchandising software solution shall provide few error rates as this is a software where analytics and proper marketing information is crucial. Therefore, the ability to readily update the Thrifty software is a key requirement. Most buyers in the United States tend to actually feel overwhelmed by too many clothes and not knowing where to find them. This can cause equal damage as not enough clothing items can in terms of marketing decisions.

**“The system shall be usable without an internet connection.”**

Not all zones in the software system will be able to reach an internet connection and some consumers may not have access to an internet connection where they place the control panel. With an increase in smartphones every year, the user may require the use of cellular data. This needs to be able to work without an internet connection to reach a larger consumer base.

### **10b. Accessibility Requirements**

**“The Thrifty software shall be readily accessible to anyone in need of apparel merchandising support and solutions”**

This will allow the Thrifty system to target bigger audiences while maintaining control in one market. For example, we are looking to support fast-fashion corporations, and in return the individual users of this market as well.

## **11. Performance**

---

### **11a. Speed and Latency Requirements**

**“The software must display statuses and feeds clearly.”**

The Thrifty apparel merchandising software solution must be prepared to readily deliver statuses in accordance to the parameters in order to maintain client Satisfaction.

## **13. Maintainability and Support**

---

### **13a. Maintenance Requirements**

**“The software of the system shall be easy to update.”**

Users are more likely to use and continue using software that is continuously progressing and making the client’s experience better.

## **14. Security**

---

### **14a. Access Requirements**

**“The system shall only allow authorized users to access the system.”**

The Thrifty apparel merchandising software solution should only allow authorized users to access the software so important information cannot be stolen.

## **14b. Integrity Requirements**

**“The product shall retain a log of all changes to the system settings.”**

The system should retain a log of all changes in the case of a failure so the system can be restored to its previous state.

## Project Issues

---

## **17. Open Issues**

---

### **17a. Open Issues**

With the increasing issue of global warming, we must decide if increasing varying weather conditions will require an increase in amount of analytics and data retrieved for the Thrifty apparel merchandising software solution.

The Thrifty apparel merchandising software should consider looking into what components make up each functional device hardware, and what software or build systems may provide longer life cycles and greater cross-platform experiences.

How can we create the most efficient yet cost effective marketing algorithm using the information available on the market?

## 24. Costs

---

Effort in person months =  $(FP/150) \times (FP^{0.4})$

Effort for RE is  $\frac{1}{3}$  of the total

Category Selection: Output triggered

Data attributes: The sensor, the current categories selected, the data, the time, and the zone

Classes: The category screen, the browsing items screen, and user profile feed

FP: 5

Buying Pattern Selection: Output triggered

Data attributes: The sensor, the current purchases, the data, the time, the current categories

Classes: The category screen, and the browsing items screen

FP: 4

Automatic Scheduling: Time triggered

Data attributes: The software, analytics, buying patterns, and the zone

Classes: The category screen, and the browsing items screen

FP: 6

Stored data

User Buying Patterns: 7 FP

Current Location Buying Patterns: 7 FP

Currently Trending Market Buying Patterns: 7FP

Non-trending Buying Patterns: 7FP

Currently Available Device: 7 FP

User Feed and Activity: 8FP

Total: 58 FP

Effort in person months =  $(58/150) \times (58^{0.4}) = 1.96$

To calculate the cost, we are using function points. The function points we have calculated are based on the attributes of each use case and the classes needed for each one. The breakdown of the function points for each use case can be seen below.

Use Case	Function Points
Category Selection	5
Buying Pattern Selection	4
Automatic Category Scheduling	6
Storing data	43

This gives us a total of 58 function points for the Thrifty e-commerce software. To calculate the effort in person months, we will use the formula:

$$\text{Effort in person months} = (\text{FP} / 150) \times (\text{FP}^{0.4})$$

Using this formula, we get approximately 1.96 person months of effort to build out the software system required for the Thrifty apparel merchandising software.