

---

CAT THINGS

---

# Frontend Requirements

---

Nicholas McHale, Andrew McCann, Susmita Awasthi,  
Manpreet Bahl, Austen Ruzicka, Luke Kazmierowicz,  
Hillman Chen

---

## **Table of Contents**

<b>Introduction</b>	<b>2</b>
<b>Product Overview</b>	<b>3</b>
<b>Users and Stakeholders</b>	<b>3</b>
User Roles	3
Customer	3
Administrator	3
Stakeholders	3
<b>Functional Requirements</b>	<b>4</b>
Use Case: Log In	4
Use Case: View Inventory	4
Functional Requirements of the Customer	5
Use Case: Check-Out Item	5
Use Case: Request Item	6
Functional Requirements of the Administrator	6
Use Case: View Transaction History	6
Use Case: View History of Item Modifications	7
Use Case: View Statistics	7
Use Case: Add New Item	8
Use Case: Check in Inventory	8
Use Case: Remove from Inventory	9
Use Case: Generate Shopping List	9
<b>Security</b>	<b>10</b>
<b>Appendix A: UI Mockups</b>	<b>11</b>

# Introduction

The purpose of this document is to serve as an agreement between the developers and the customers on what the frontend will do. The target audiences for this document are the project stakeholders. Please review this document and comment on anything that may need clarification.

## Product Overview

*THINGS* is an inventory control system for the C.A.T. Its purpose is to track the on-hand inventory of equipment and consumable products. It will allow for easier item management by highlighting usage trends and giving alerts when products fall below a predefined threshold. Specifically, the front-end will allow both general users and administrator to interact with the inventory: general users can check out or request items while administrators can manage the inventory.

## Users and Stakeholders

### User Roles

There will be two types of user roles for the front-end. The roles are Customer and Administrator.

#### **1) Customer**

- a) A customer will typically be a C.A.T. employee or volunteer. The customer will interact with the front-end to check out inventory or product.

#### **2) Administrator**

- a) An administrator will be Theodore or another C.A.T. employee who is responsible for shopping and inventory management. The administrator will use the system to track inventory and keep products in stock. The system will help the administrator by triggering low stock warnings, and tracking usage trends.

### Stakeholders

- 1) Customer:** Computer Action Team (C.A.T.)
- 2) Project Sponsor:** Theodore Mason & Aaron Goodstein
- 3) Development Team:** Nick McHale, Hillman Chen, Susmita Awasthi, Austen Ruzicka, Luke Kazmierowicz, Andrew McCann, Manpreet Bahl
- 4) Capstone Sponsor:** Bart Massey

# Functional Requirements

The following two requirements are independent of whether the user is the customer or the administrator.

## Use Case: Log In

**Brief Description:** There will be two accounts: one for customers and one for administrators. The account for customers can only checkout or request items while the administrator can do everything.

**Assumptions:**

- 1) Credentials are present in the database.

**Step-by-step:**

1. User indicates they would like to log in.
2. User enters account name into account name field.
3. User enters password

**Post-Condition:** The user is logged in with either customer or administrator privileges.

## Use Case: View Inventory

**Brief Description:** This will display a list of all the items in the inventory and the quantity in stock.

**Assumptions:**

1. Inventory exists (can be empty or contain list of items divided into categories such as name, quantity, price, tags, and threshold).

**Triggers:**

1. Call the view inventory API call is made to get a up to date list of the inventory for the user to choose items to checkout or check-in.

**Step by Step:**

1. System gets the inventory list and displays it to the user.
2. User can filter the inventory using the following options:
  - a. Categories:
    - i. Name
    - ii. Tags
2. System outputs the list of items based upon the filter options selected.

**Post-Condition:** Database Inventory remains unchanged.

# Functional Requirements of the Customer

## **Use Case: Check-Out Item**

**Brief Description:** The Customer takes some things from the supplies cabinet and enters what they took into the system. See figures 1-4

### **Assumptions:**

- 1) Database has no errors in its data.
- 2) Item is present in supply cabinet.

### **Step-By-Step:**

1. The system displays a welcome screen that contains a field for the user to enter their name.(Figure 1)
2. The user types in their name.
3. The system displays a scrolling list of all items in the database sorted in alphabetical order (Figure 2).
4. The user finds the item to be checked out in the list.
  - a. The user clicks the Add button.
  - b. If item is not in cart, then it is added to cart. Otherwise, the item is already in cart so the quantity to check out is increased by 1.
5. The user clicks the “Checkout” button to proceed with checkout. The user also has the option to remove item(s) from the cart by selecting the items and clicking on the “Remove” button.
6. The system displays a message depending on whether it was a successful transaction or not.
  - a. Successful: The system will display a message indicating that there were no errors.
  - b. Failure: The system will display a list of items that were not able to be checked out.
7. The user clicks the “Done” button.
8. The system reloads the page for the next user to use.

### **Post-Condition:**

- 1) The transaction details are stored in the database.
- 2) The database reflects the changes in quantity.
- 3) If an item’s quantity drops below threshold, an alert is sent.

### Use Case: Request Item

<b>Brief Description:</b> The customer may submit a request for items they are not currently stocked. This request generates an email to an administrator (See Figure 5)
<b>Assumptions:</b> 1) The item being requested is not currently an item being stocked.
<b>Triggers:</b> 1) The user clicks on “Request” tab on navigation bar, fills out the form, and clicks on “Submit”.
<b>Step by Step:</b> 1) The user is prompted to fill out a form with the following: a. Item Name b. Quantity c. Item Description d. Additional information. 2) The user submits the form. 3) An email is generated to the admin.
<b>Post-Conditions:</b> 1) The database is unchanged. 2) An email with the users request is sent to an administrator.

## Functional Requirements of the Administrator

### Use Case: View Transaction History

<b>Brief Description:</b> The view transaction history use case will display a list of the 15 most recent transactions with an option to view the previous 15 and so on.
<b>Assumptions:</b> Database can track recent transactions. Database has a list of recent transactions. Admin is authenticated. Database values are correct.
<b>Step-by-step:</b> 1. Beginning at the ‘admin landing page,’ the admin will click on the button that corresponds with the “View Transaction History” case. 2. The system will display the 15 most recent transactions, including check-outs, and check-ins. a. Entries will link to item info page.
<b>Post-Condition:</b> Database remains unchanged.

### **Use Case: View History of Item Modifications**

**Brief Description:** The View Item Modifications history use case will display a list of the 15 most recently altered items.

**Assumptions:**

Database can track all modifications to the items being tracked in inventory.

Admin is authenticated.

Database values are correct.

**Step-by-step:**

3. Beginning at the 'admin landing page,' the admin will click on the button that corresponds with the "View Item Modifications History" case.
4. The system will display the 15 most recent modifications to the list of items. This includes:
  - a. Items added
  - b. Items removed
  - c. Items modified

**Post-Condition:** Database remains unchanged.

### **Use Case: View Statistics**

**Brief Description:** This use case is related to View transaction history, but it shows us item history in a graphical way. This use case will only be feasible in the front end application.

**Pre-Conditions:** Logged in as administrator

**Assumptions:**

- 1) Inventory exists (can be empty or contain list of items divided into categories such as name, quantity, price, tags, and threshold).
- 2) By default, all items in inventory are displayed in alphabetical order of item name by the system.

**Triggers:**

- 1) The administrator selects the "View Statistics" option

**Step-by-Step:**

- 1) The administrator selects an item and optionally a time frame
- 2) A graph of Item Quantity vs Time will be returned
- 3) A graph of Average Daily usage rates by day of week.

**Post-Conditions:**

- 1) The database remains unchanged
- 2) The user is given a graphical representation of item history.

### **Use Case: Add New Item**

<b>Brief Description:</b> This use case will outline the process for adding a new item to the database
<b>Pre-Conditions:</b> Logged in as administrator
<b>Assumptions:</b> <ol style="list-style-type: none"><li>1. Inventory exists (can be empty or contain list of items divided into categories such as name, quantity, price, tags, and threshold).</li><li>2. The default threshold is 0 if the administrator doesn't specify the threshold when adding item to inventory.</li></ol>
<b>Triggers:</b> <ol style="list-style-type: none"><li>1) Administrator selects "Add New Item "</li></ol>
<b>Step-by-Step:</b> <ol style="list-style-type: none"><li>1) The following information is entered:<ol style="list-style-type: none"><li>a. Name</li><li>b. Description</li><li>c. Quantity</li><li>d. Price</li><li>e. Tags</li><li>f. Threshold</li></ol></li><li>2) System ensures that the item does not exist in the inventory.<ol style="list-style-type: none"><li>a. If it does exist, inform the administrator that the item already exists as well as return the item information.</li></ol></li><li>3) Item is added to the inventory and a confirmation message is displayed.</li></ol>
<b>Post-Condition:</b> The new Item is added to the inventory.

### **Use Case: Check in Inventory**

<b>Brief Description:</b> This use case will outline the process for an administrator to check-in recently purchased goods into inventory
<b>Pre-Conditions:</b> Logged in as administrator
<b>Assumption:</b> <ol style="list-style-type: none"><li>1) The item already exists in the inventory</li></ol>
<b>Trigger:</b> <ol style="list-style-type: none"><li>1) Administrator selects "Check in Inventory"</li></ol>
<b>Step-by-Step:</b>



<ol style="list-style-type: none"> <li>1) The administrator selects from the inventory list the item(s) to be checked in.</li> <li>2) The administrator specifies the number to be checked in for each item.</li> <li>3) The administrator clicks on “Checkin” to check in the items. The system will display a message if the transaction(s) were successful or not. <ol style="list-style-type: none"> <li>a. Successful: A message will indicate that all items were checked in.</li> <li>b. Failure: A list of items that were not able to be checked in will be displayed.</li> </ol> </li> <li>4) System reloads the page for the next transaction/user.</li> </ol>
<b>Post-Condition:</b> The Item quantity is updated in the inventory.

### **Use Case: Remove from Inventory**

<b>Brief Description:</b> This use case will document the process to remove an item from the inventory tracking
<b>Pre-Conditions:</b> <ol style="list-style-type: none"> <li>1) Logged in as administrator</li> </ol>
<b>Assumptions:</b> <ol style="list-style-type: none"> <li>1) Inventory exists (can be empty or contain list of items divided into categories such as name, quantity, price, tags, and threshold).</li> </ol>
<b>Triggers:</b> <ol style="list-style-type: none"> <li>1) Administrator selects “Remove Item from Inventory”</li> </ol>
<b>Step-by-Step:</b> <ol style="list-style-type: none"> <li>1) The following information about the item is entered: <ol style="list-style-type: none"> <li>a. Item Name</li> </ol> </li> <li>2) System ensures that the inventory is not empty and searches for the item. <ol style="list-style-type: none"> <li>a. If not found, display a message stating the item was not found.</li> <li>b. If found, delete the item, and display a confirmation message to the administrator.</li> </ol> </li> <li>3) Administrator exits the screen</li> </ol>
<b>Post-Condition:</b> If found, the item is removed from inventory.

### **Use Case: Generate Shopping List**

<b>Brief Description:</b> The Generate Shopping List use case will create a list of items based on the count of items remaining, in relation to that item’s threshold.
<b>Assumptions:</b> <ol style="list-style-type: none"> <li>1) Database has correct entries.</li> <li>2) Admin has been authenticated.</li> </ol>

**Step-by-Step:**

1. Beginning at the “admin landing page”, the admin will click on “Shopping List” to navigate to that page.
2. Display all items that are at, or below the threshold in a printable format in a new tab.
  - a. Items will be displayed with their name, threshold, and current count.

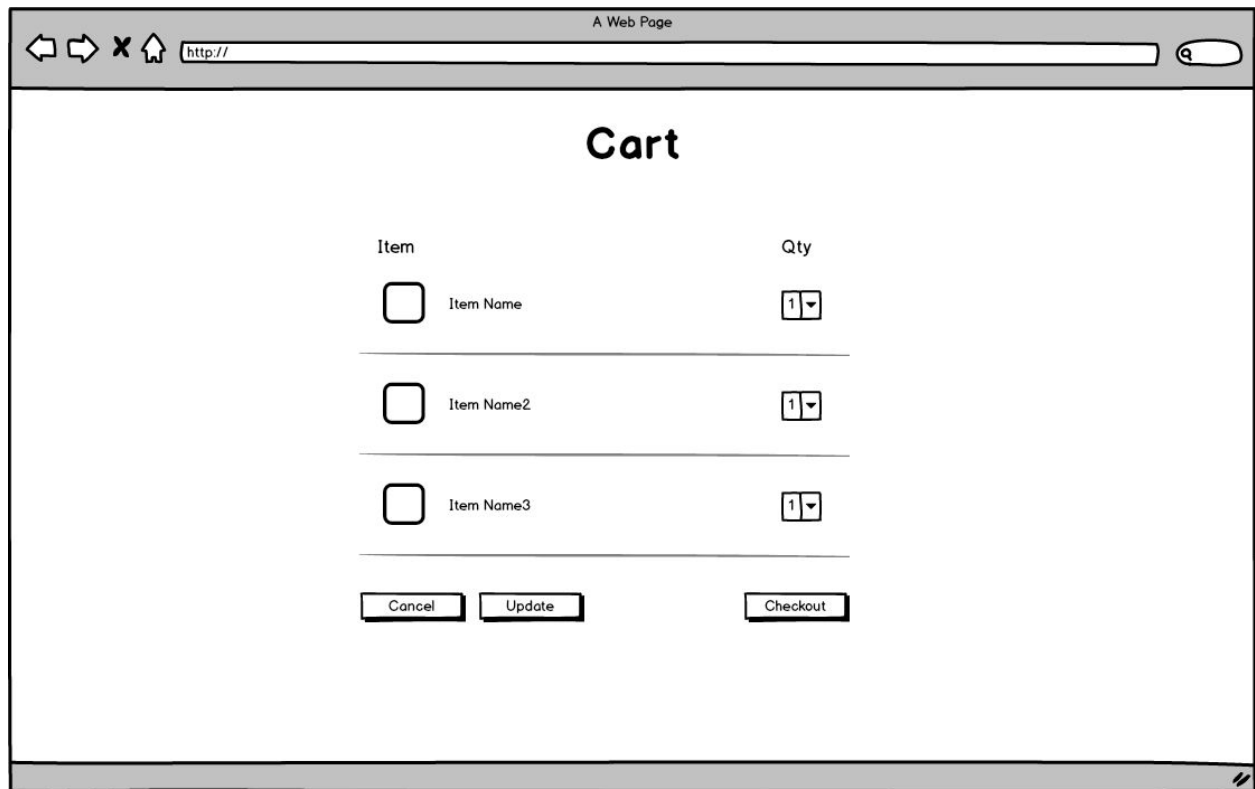
**Post-Condition:**

- 1) A shopping list report is generated for the user
- 2) The database remains unchanged

## Security

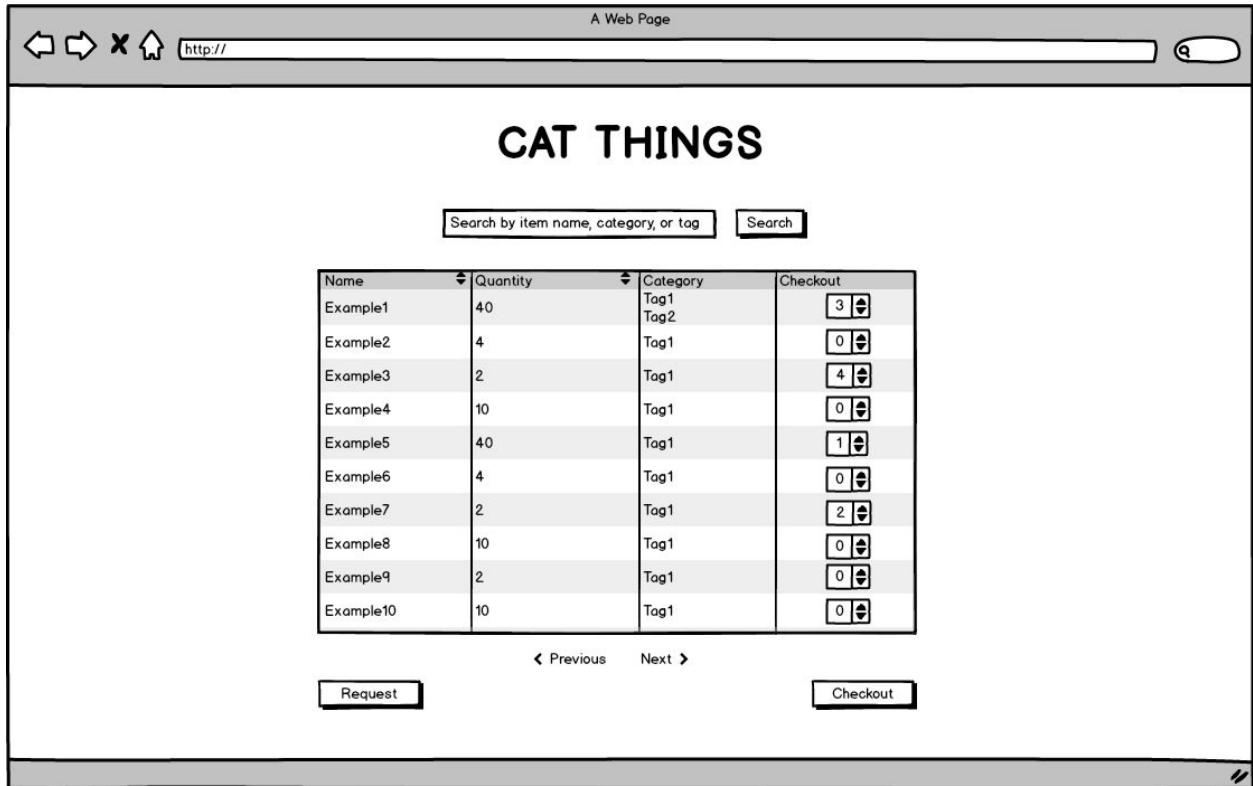
We will assume that no confidential or secret information is stored in this database. Therefore, the biggest risk to security for this project is data integrity. This is not because of any negative consequences from anyone being able to view the data, but rather, we want to ensure our stock quantities are accurate and are not being manipulated by a third party, either maliciously or inadvertently.

# Appendix A: UI Mockups



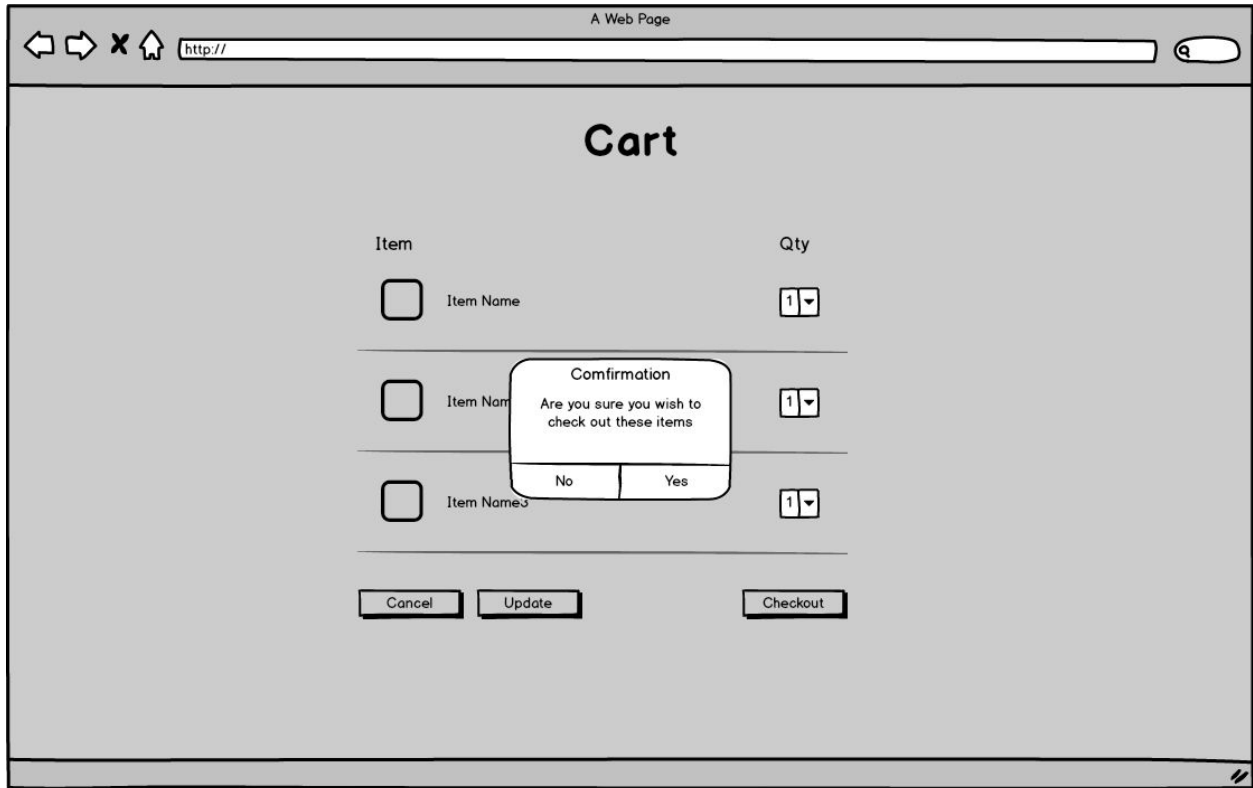
Created with Balsamiq - [www.balsamiq.com](http://www.balsamiq.com).

Figure 1



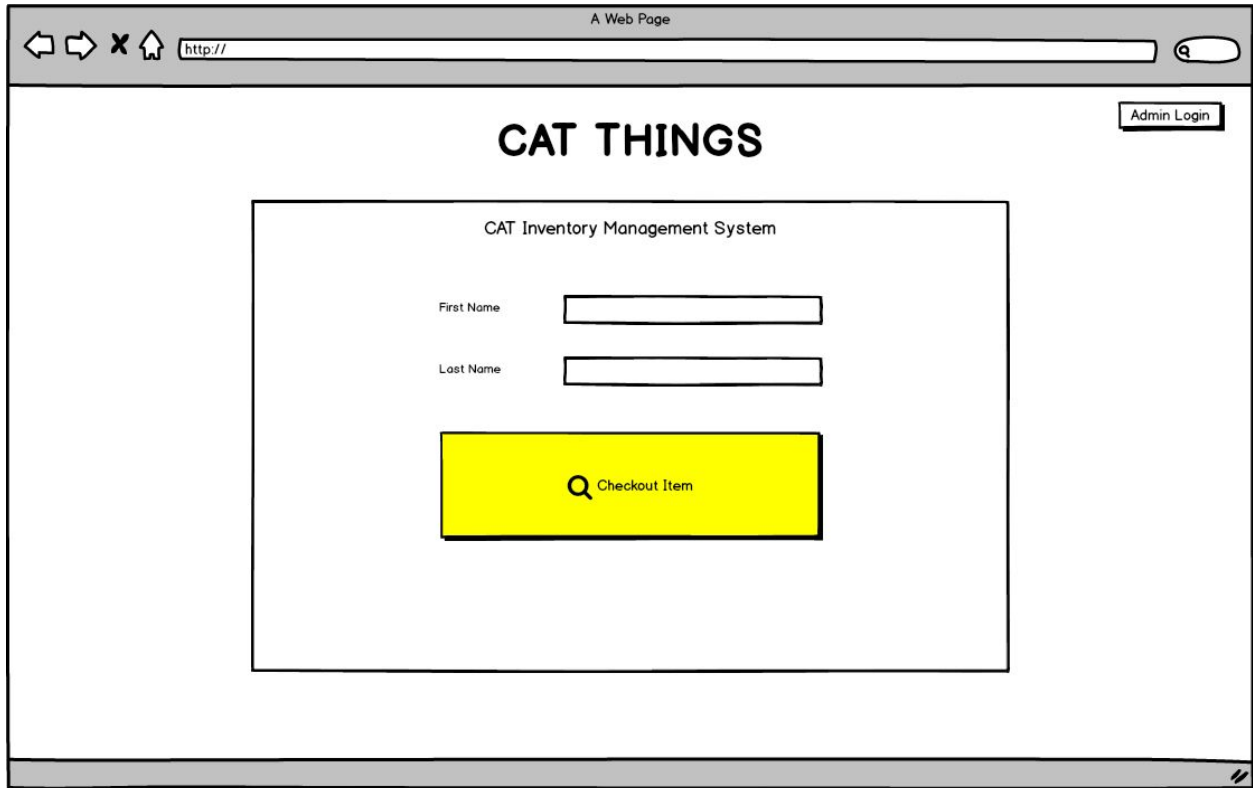
Created with Balsamiq - [www.balsamiq.com](http://www.balsamiq.com).

Figure 2



Created with Balsamiq - [www.balsamiq.com](http://www.balsamiq.com).

Figure 3



Created with Balsamiq - [www.balsamiq.com](http://www.balsamiq.com).


Figure 4

A Web Page


http://

## Request An Item

Details

Date  

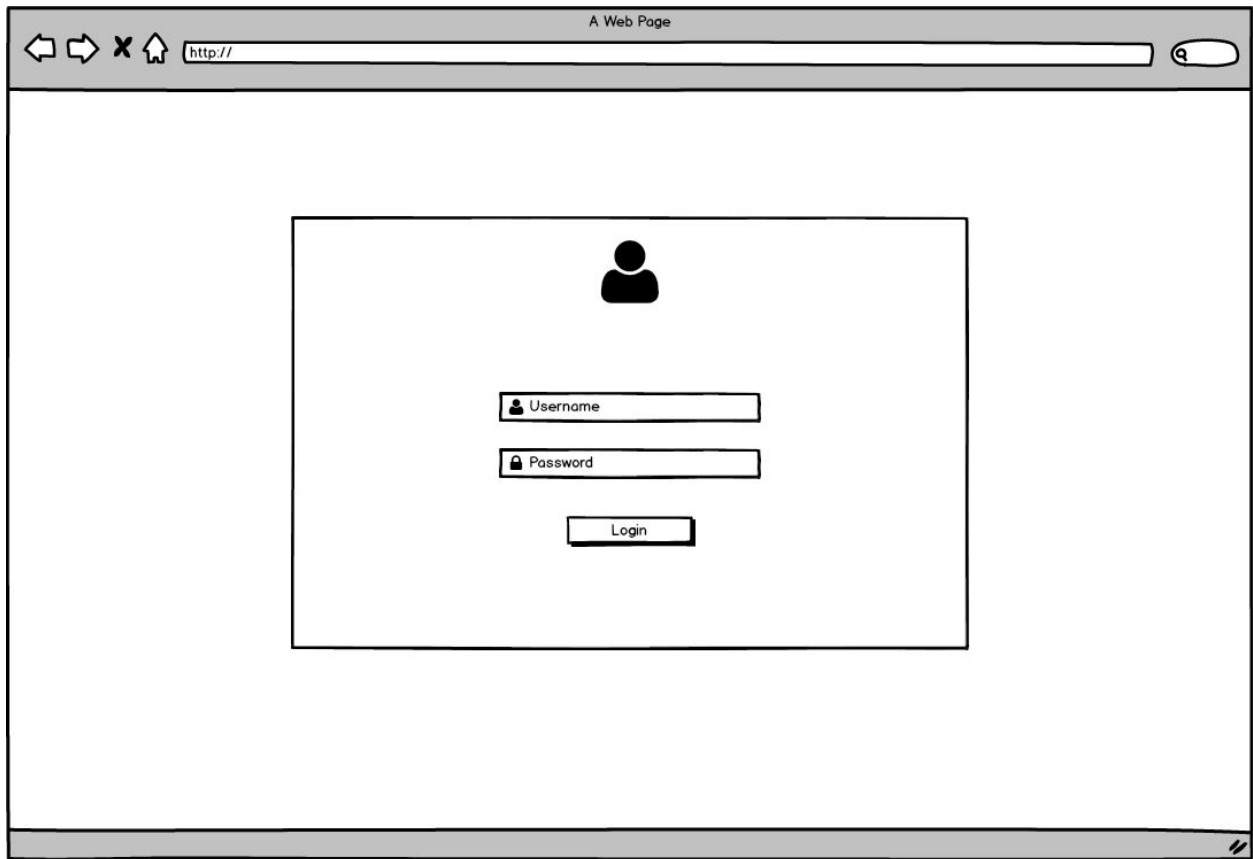
Item Name

Quantity  

Description

Created with Balsamiq - [www.balsamiq.com](http://www.balsamiq.com).

Figure 5



Created with Balsamiq - [www.balsamiq.com](http://www.balsamiq.com).

Figure 6



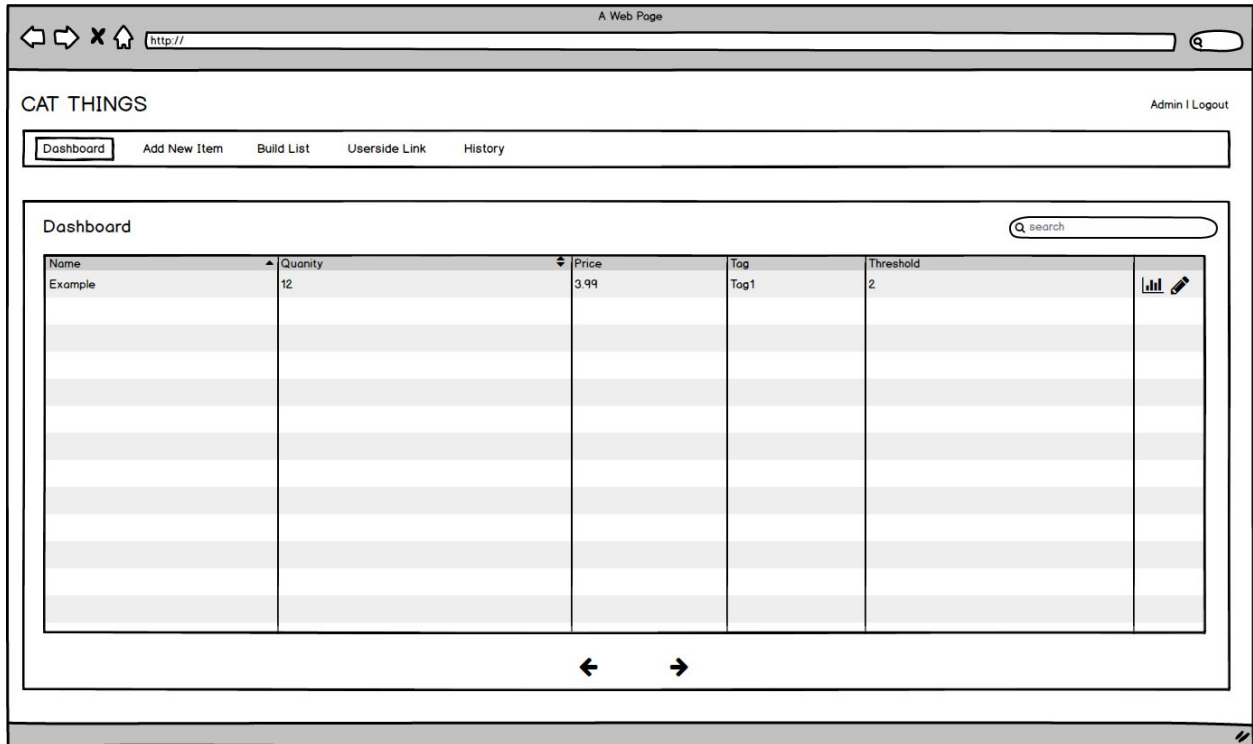


Figure 7

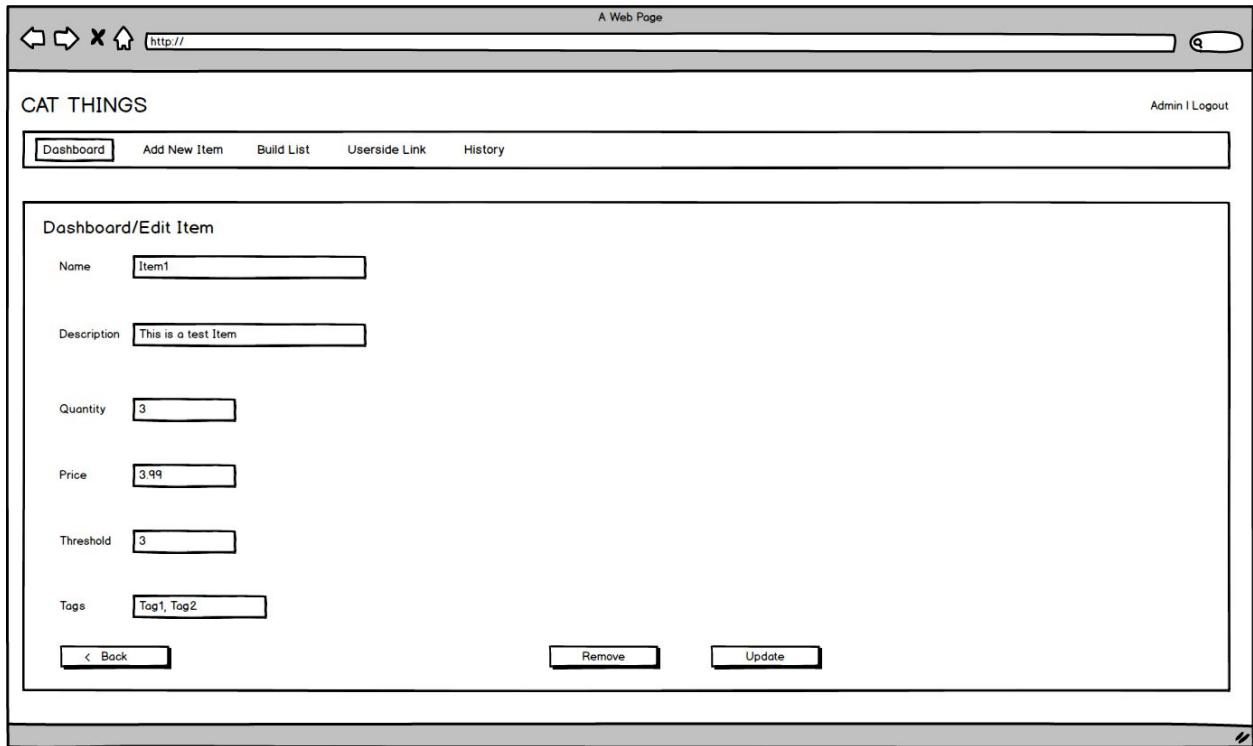


Figure 8

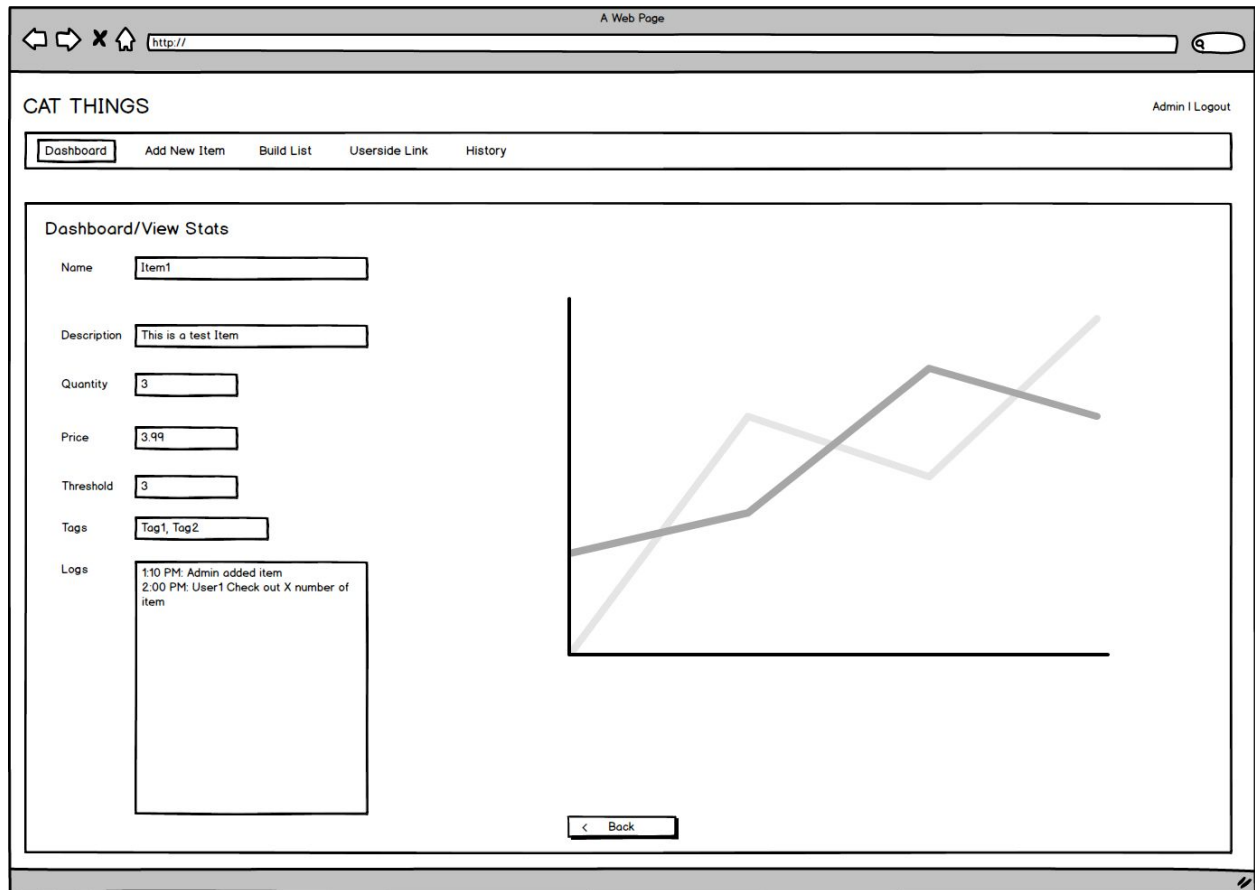


Figure 9

A Web Page

http://

CAT THINGS

Admin | Logout

Dashboard

Add New Item

Build List

Userside Link

History

Add New Item

Name

Description

Quantity

Price

Threshold

Tags

Add new Item

Figure 10