CAT THINGS

Frontend Design

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

Table of Contents

Introduction

THINGS is a web-based inventory control system for the C.A.T. The front-end of the system is responsible for interaction with the user and admin to input and check out items from the inventory system. The front-end is also responsible for packaging this information and passing it along to the API. The system will help the C.A.T. to manage their inventory and observe its usage trends. The system is anticipated to be used principally by C.A.T. associates as users and administrators.

Goals

The primary goal of this project is to provide the C.A.T. with a system to help them manage their inventory. The front-end is intended to be the interface between the system and the user or administrator. The front-end should:

1. Provide an easy to use, functional user interface.
2. Allow a user to check-out an item.
3. Allow an administrator to directly manage the inventory.
4. Package information from the user and pass it on to the API reliably.
5. Display inventory information.
6. Generate a shopping list to the user.
7. Generate statistics about the system.
8. Be stable and predictable.

Architecture

The system will be web-based. The front-end will be a series of web pages to service user's requests. The front-end will take requests from the user, and pass them along to the API which will access the database. All input will be from the user which will encompass requests to access the database to modify or manage the inventory. Another form of input will be from the API itself as an answer to the requests sent from the front-end. The outputs from the front end will be the user's requests. The system will be accessed from a computer in the C.A.T.'s office via web. There should only ever be one instance at a time of the front-end, unless it is accessed through more than one computers at a time. The front-end will run on Angular 1 and asynchronous javascript as well as the C.A.T.'s infrastructure.

Design

The front-end is developed using AngularJS, angular directives for Bootstrap, HTML and CSS. Each HTML page has its own controller JS file that controls aspects of that page from displaying the inventory in a table to submitting forms. In addition to the

controllers, there's another set of JS files called services which maintain data that multiple controllers can use. This includes the list of inventory, checkout cart, check-in table, and an object used to communicate with the API called thingsAPI.

When the front-end is accessed, the first page that the user will encounter is the login screen where the user will enter the appropriate credentials. This will make an API call (using the thingsAPI object) to validate the credentials and if successful, load the homepage with the appropriate privileges:

1. Customer can only checkout and/or request items.
2. Administrator can perform all tasks that the customer can as well as check-in items, add new items, view transaction history, view statistics, and generate shopping list for items at or below their thresholds.

The front-end is expected to be only be used by C.A.T. associates as either users of the system or as administrators and will require no experience or knowledge to operate. There should only be one user using the front-end at any given time.

Customer

The customer has access to the either checking out item(s) or making request for items that don't exist in the inventory.

Checkout

The checkout page allows the user to search and select item(s) from the inventory list that they wish to checkout. There will be an "Add" button next to each row in the inventory list which will add the item to the Cart table or increment the quantity to checkout by 1 if the item is already in the cart. The user then can proceed to checking out the items by clicking on the Checkout button. This will result in making multiple API calls, one for each item to checkout, till all items have been checked out successfully. If any errors occur, the front-end will display it so with the list of item(s) that failed to checkout.

Request

The request page allows the user to request items that are not available in the inventory. The user enters the item name, quantity needed, item description, and any other additional information that would help the administrator obtain/purchase the correct item. Once the user submits this form, an email will be generated to the CAT administrator.

Administrator

The administrator can perform all actions that the customer in addition to having more control of the inventory. The administrator can check in items, add new items to the

inventory, view statistics and generate a shopping list to stock up on items that are out of stock or running low.

Check-in

The checkin page allows the administrator to search and select item(s) from the inventory list that they wish to check-in. This page is similar to the checkout page except that a checkbox replaces the “Add” button and a different service is used to maintain the check-in table. When the user clicks on the checkbox, it’ll add that specific item to the check-in table and then the user sets the quantity that is being checked-in. The user then can proceed to checking-in the selected items which will result in a success or failure message being displayed in the same manner as the checkout page.

Add New Item

This page allows the administrator to add a new item to the inventory by submitting a form. The administrator needs to specify the following information about the item:

1. Item name
2. Price
3. Threshold
4. Item description
5. Initial quantity
6. Tags

Once the form has been filled out and submitted, an API call is made which adds the new item to the database.

Shopping List

This page allows the administrator to view/print the list of items that are below their threshold or out of stock. When the page is loaded, an API call is made to get the list of items and then is displayed in a nice format.

View Statistics

This page allows the administrator to view statistics about items and their transactions such as checkouts per day and average amount consumed by week. It also allows the administrator to display graphs for a selected item.

View Transaction History

This page allows the administrator to view transaction history and can select from the following 4 options:

1. **Number of Recent Transactions:** The administrator will enter a number and the front-end will display the most recent transactions up to the number entered.
2. **Item Name:** The administrator will enter an item name and number of transactions to display and the front-end will display the transactions matching that item name.

It’s important to note that different items can share the same name and all items with such names will be displayed.

3. **Item Tags:** The administrator can enter tags of interest and the number of transactions and the front-end will display transactions matching the tags up to the number of transactions specified.
4. **Date Range:** The administrator can enter a date range and the front-end will display all transactions within the specified the range.