Home Re:Stock

Item detection and usage measurement

Blake Reed, Emad Siddiqui, Jan Timpe

I. Introduction

Being generally busy with everyday tasks, many people simply do not notice when they are running out of common household items or groceries. With the average person spending 53 hours per year and making an average of 1.5 store trips per week grocery shopping, it is clear that the market demands a solution for tracking product usage and providing a simplistic way to be notified of and reorder empty items. Time and money are wasted when a consumer cannot properly track product supply and usage.

A solution to this problem is a smart home restocking system; a product resembling a surface with embedded pressure sensors and camera components that identifies and tracks usage of common household and grocery items, prompting the user to reorder when supplies are low. Additional sensors and environmental controllers, though outside the scope of this project, may be added to increase energy efficiency (in the case of a refrigerator or freezer) and keep products at the correct temperature, humidity, etc. to cut down on food waste..

II. Related Work

Some existing solutions are Prediction Algorithms, Amazon Dash Buttons, Amazon Auto Reorder, and Scan-n-Track systems. What makes our solution unique is that it works with definite values unlike prediction algorithms, which in the end are just predictions. Amazon Dash Buttons are limited to one retailer/one product and are used on a need-deliver basis. Our product is versatile, it piggybacks on multiple retailers and is able to re-order multiple products. The Scan-n-Track systems are built for retail while our product is targeting average consumers. Our product makes accurate orders where the rest of these competitors fail to.

III. Hardware Design

Prototype design for Home Re:Stock included consisted of a camera, a set of load cells beneath a solid platform, and a Raspberry Pi acting as a client/microcontroller and a web server. The camera and load cells work together through the client-side script to determine the existence and status of items on the platform.

Several solutions for both the camera and weight sensors were tested. The provided Raspberry Pi camera component had multiple issues with focus on the platform. Eventually, it was replaced by a webcam with a manual focus to ensure that items could rest on the platform inches from the camera to provide clearer images of barcodes, both in terms of camera quality and size of the barcode in the image. A standard definition webcam, which had multiple reliability issues, including frequent crashes and disconnections, and poorer image quality overall, was then replaced with an HD webcam, also with a manual focus, which allowed for better quality images and greater reliability.

The initial product design included Velostat, an electrically conductive material, to measure changes in weight on the platform. After much research, some standard Arduino/Raspberry Pi load cells, along with an HX711 amplifier, was chosen to replace the Velostat in order to provide accurate weight measurements, rather than just detecting changes in pressure on the platform. The load cells and amplifier, however, turned out to be wholly unreliable, returning inconsistent (and blatantly incorrect) readings. After testing with three Raspberry Pi controllers, no configuration issue could be found with the wiring of the individual cells or connectivity between the amplifier and the Raspberry Pi, thus some undescribed hardware issue is suspected. A final attempt using a Wii Fit Balance Board, connected to the Raspberry Pi via Bluetooth, yielded incorrect, although consistent, readings, and was not sensitive enough to detect small weight changes. Ultimately accurate readings are needed to implement the client side controller so the weight sensing functionality of Home Re:Stock remains unimplemented.

IV. Application Design

The application implementation of Home Re:Stock included a Flask web application and a client side script written in Python. The client script, meant to provide connectivity to the camera and weight sensors, both detects and monitors the weight of items placed on the described platform. The camera, using OpenCV and Zbar, actively detects and scans barcodes. When an item is detected on the platform, a weight measurement is obtained through the load cells, and a POST request is made to the Flask application, including the item's UPC barcode and measured weight.

The Flask application selects the item from a database by the barcode and adds an item history including the weight and current timestamp. If the item does not exist in the local database, an API request is made to the Walmart Open API, searching via UPC barcode, and collecting information and inserting it to the local database. Data collected from the Walmart API includes the product name, URL for the item's listing online, URL for adding the product directly to the user's online cart, and the item's size attribute if it is listed.

Additionally, each product entry in the database includes an "empty" flag, to be turned to true if the client script detects that an item falls below the emptiness threshold (currently set to 15% of the item's initial weight as reported by the Walmart API).

The final layer of Home Re:Stock is an Android application, designed to connect to the Flask server and display a list of the products, their weight histories since being added to the platform, and the empty status of the item. The data from the Flask server is retrieved asynchronously from the Flask REST endpoints and stored in models which make the JSON return data more manageable within the application. The home activity's listview is nested within a SwipeRefreshLayout in order to allow the user to manually refresh their shopping list while using the application. Additionally, the user is provided with a "Reorder" button to allow them to easily add the item directly to their Walmart online cart which redirects the user to the "add_to_cart_url" provided by the Walmart API upon scanning an item. This allows the user to keep a shopping list feel while effectively keeping track of their Walmart shopping cart. For each item, a usage history graph is shown, filterable by day, week, month, and year time frames. This utilizes the GraphView android library to embed a graphical representation of weight and date data into the activity. The filter is implemented via a spinner dropdown option and checks date ranges of usage history in order to display relevant data.

V. Results

Overall, our prototype was able to solve the problem and able to fulfill the goals we had in mind at a very basic level. We were not able to get a weight sensor correctly working, the ordered velostat and load cells with HX711 amplifier were nowhere near accurate/consistent enough. A failed attempt at an alternative solution was using a Wii Fit Balance Board, it's measurements were consistent but not accurate and it was nowhere near sensitive enough as we needed it to be. Although the prototype was not perfect (no weight sensor), it was able to correctly identify products, keep track of product usage on a daily/weekly/monthly/yearly basis and prompt reorder upon detection of low stock using the mocked weights. An improvement could have been made to the UX/UI design of the android application in order to make it easier to navigate through.

VI. Future Work

Home Re:Stock so far is just a base prototype of our original idea, a start to a much greater picture with many more possibilities and features to come. In the future, the product will consist of a hardware component, the physical surface/mat on which items will be placed. The initial design for the surface includes a glass/acrylic base in which camera components will be housed. The cameras will be arranged such that the array of cameras can capture products placed in any position on the mat. The glass housing will allow barcodes to be detected from the bottoms and undersides of each product. The prototype as of right now only handles one product, but the goal is to detect as many products as there are placed on the surface. Much experimentation must be done to correctly configure the camera arrangement inside the glass housing in order to maximize the area that can be covered by barcode scanning. Scanning barcodes from a direct angle is trivial using common Python libraries (zbar), but may be more difficult with multiple products arranged in various different angles. Weight/Pressure sensors will be embedded in the glass surface as well, allowing the weights of individual products to be captured and recorded. Additionally, weight sensors must be configured to accurately weigh a single item (among potentially many). A solution to this is capturing multiple readings from each sensor and camera throughout the user's interaction with the products on the surface. This will allow the application to intelligently deduce which items are being used over the course of the interaction and assign values accordingly. More work must be done to integrate vision APIs ato detect items that are not packaged, or unconventionally packaged, and may not contain a standard, readable barcode. The mobile app needs to add more vendors/retailers other than just Walmart, have it set to where it will suggest the lowest priced item with also factoring in pickup/delivery times. Any retailer with an online ordering system would eventually be supported, and thus their would be no limiting products to one supplier. A basic android application has been implemented but once more features have been added and the UI has been improved, the iOS application would also be developed.

Once the Home Re:Stock surface has had all it's features implemented, there are many possibilities to where it can go from there. The surface could be transitioned into being a smart fridge, this technology can be extended to include additional sensors, such as temperature and humidity sensors, and vision API's (such as those provided by ClarifAI and Google Vision) to detect grocery products and automatically adjust the environment to maximize preservation and shelf life of those items. Another possibility is transitioning it into a larger/more practical solution of a surface, a Home Re:Stock countertop.

VII. Appendix

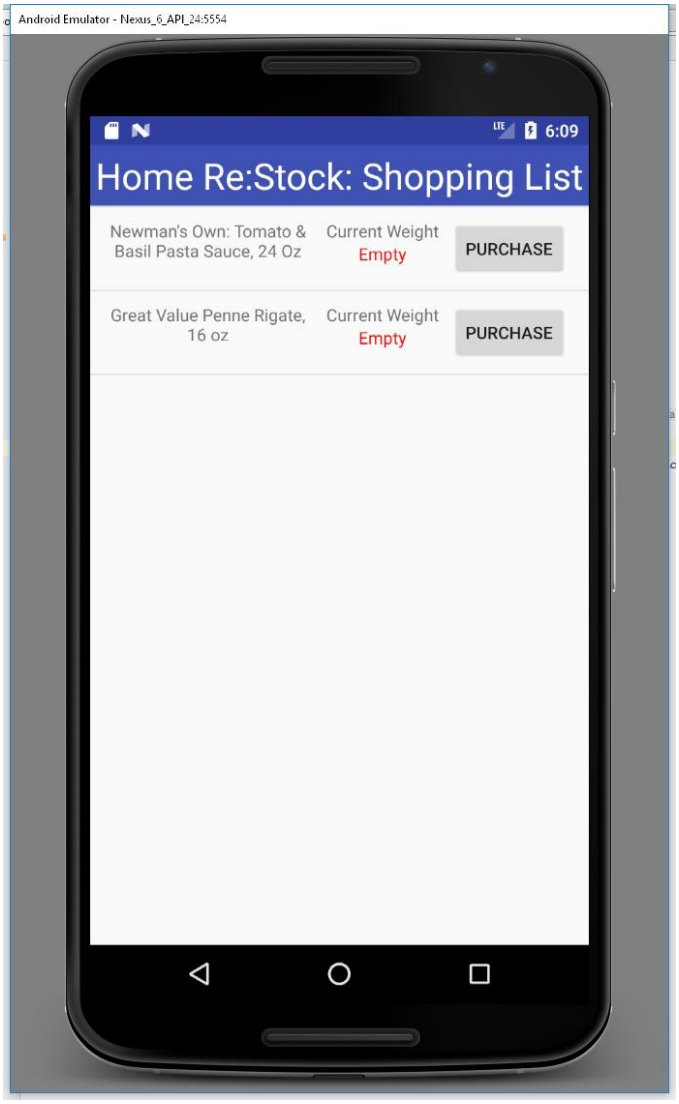

Figure 1: Home Re:Stock mobile home page and shopping list view. Contains each product scanned by the user along with a purchase option.

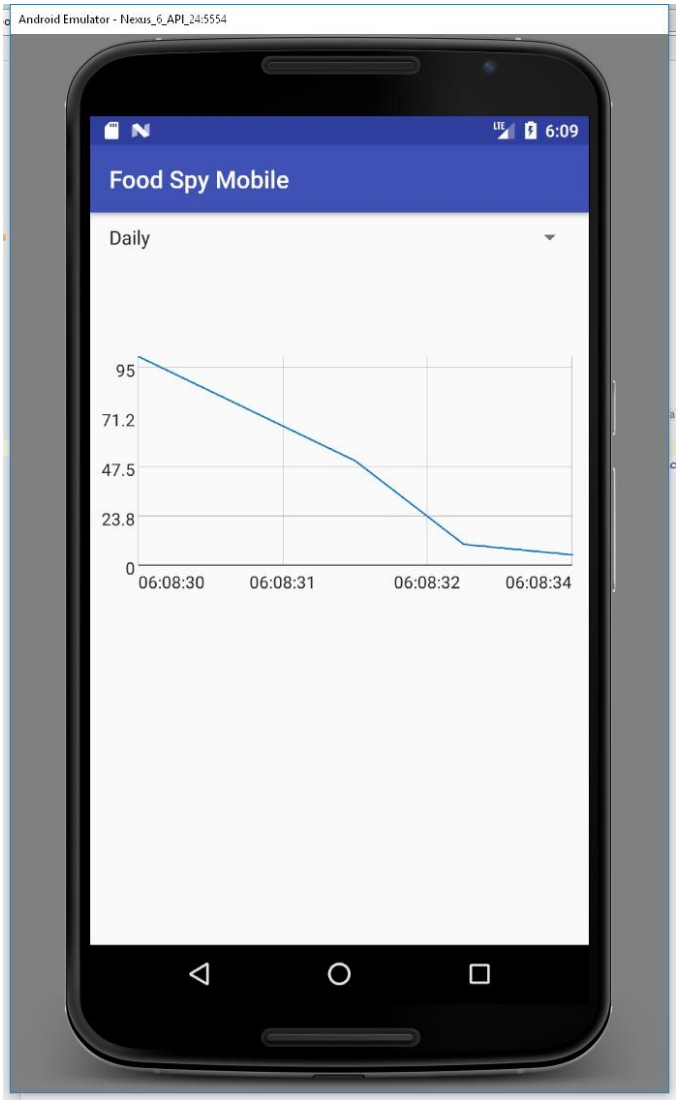

Figure 2: Filterable GraphView of food item usage in order to display to the user rate of consumption and prompt reordering of items.

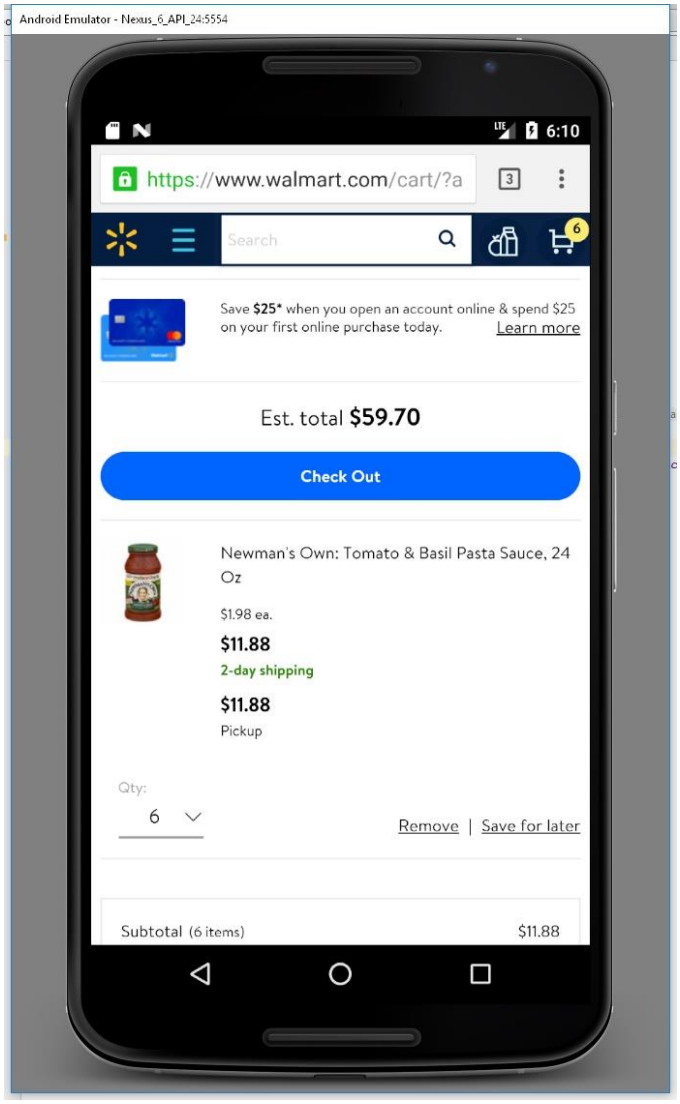

Figure 3: Add to cart url linked to purchase button which keeps a comprehensive shopping cart via Walmart online shopping.