Extended Problem Statement and Requirements Specification

Inventory Management System

Authors: Bradley Griffee, Brian Sizemore, Mark Morrison

Table of Contents

Executive Summary

The goal of our project is to create both an inventory management system (IMS), and a mobile application that will allow users to easily interface with our system. First, we will build a system which stores information about items and allows users to easily search, retrieve, and modify the information it contains. Secondly, we will design and build a mobile application that can be used with the IMS. The application will allow a user to easily add new items to the system, by storing information about the object and by uploading photos of important information on the object, such as the serial number. Together the application and the IMS will form a robust system.

Introduction

The goal for our team this semester is to create a fully functional inventory management system (IMS). The IMS will be used in an effort to simplify the logistics of managing non-traditional inventories. We believe current methods of inventory management are lacking in their ability to manage large sets of unique items in a clean and easy to use fashion.

Current competitive options in the market have the issue of being overly complicated and over developed. Losing sight of the end-user in the circumstances where products such as this are most crucial to the development of the business. For example, one such inventory system out there is Bizslate [1]. The marketability of this product, and likely the driving factor for its cost, is the “features” it provides, which claim to be able to make your business more profitable. However, the average small business owner doesn’t know what half of the charts and graphs these “features” generate even mean.

This is just one of the topics our product will tackle: simplicity. The average mom and pop shop should be able to pick up their phone and intuitively be able to track their products in a systematic and controlled manner. No complicated, confusing interfaces, and no unnecessary clutter. Manage inventory, make money. Simplicity, controlled.

Extended Problem Statement

Obviously, the point of all businesses' is to make money, and in most cases this is done by providing an inventory of items that people want to purchase. Modern technology has made huge strides in helping to manage the logistics of most companies by providing common ways to track and manage inventory. For example, in a grocery store, whenever items are purchased they come with a UPC (unique product code). This allows for the store receiving the items to easily scan the items as they come in and all of the important information about that item is then added to their system.

However, in the case of many newer companies, inventory is less regulated and less standardized, so there may not be a UPC already available for tracking the item. Our inventory management system will aim to remedy this situation by creating a way for modern users to utilize the technology already available on their smartphone to seamlessly integrate the inventory management system into their already operating business.

For many years' owners of businesses that kept non-traditional stock – e.g. not already labeled and packaged – have struggled with the ability to accurately keep track of expanding businesses. But this issue is not only for small business', but rather it impacts almost any organization that has any kind of property they wish to keep track of. A perfect example of such a case is the West Virginia State Police: they have items – cars, radars, laptops, etc, scattered all over the state which may belong to dozens of different officers throughout their lifecycle. But uniquely identifying all of these items and keeping track of where they are and where they have been is an entirely different beast than they are used to dealing with. A generalized inventory management system can provide the ability to give all of these items unique tracking codes or product identifiers based on already existing unique properties, like serial numbers and VIN numbers. One system which can handle a large variety of items would be beneficial to any organization with items they want to keep track of.

This is where our product aims to step in. By leveraging the already exploding access to smart devices, such as smartphones, we can implement management systems with minimal expenses or changes to the existing architecture of business'. Since minimizing costs is a key component of running a successful small business, the principle of minimizing costs and maximizing ease of use will bring in endless stakeholders and potential customers.

Requirements Specification

This section contains the functions which should be available to users, the engineering requirements for a product to be successful, and the marketing requirements to make sure that it is something people would actually buy in to.

Functional Requirements

a. **Add new item to inventory**
   - The user should be able to add items into the database which will accept a variety of information for tracking purposes. Adding items should generate a unique identifier code for use in the database and provide the ability to put that code onto the item (via printing a QR code or encoding an NFC tag).

b. **Delete item from inventory**
   - The application should provide a way to remove items from inventory at will manually alongside the automated updating process (see export items below).

c. **Modify item in inventory**
   - The user should be able edit items that are already in the database in order to change anything about them in the database.

d. **Search current inventory**
   - The application should provide the ability to browse the current database based upon several distinguishing factors such as unique code, describing factors, serial number, manufacturer, and when the item was added to the inventory.

e. **Export items from database**
   - The interface should provide an easy (preferably automated) way of adding items to, and removing them from the database based upon a typical export format that an e-commerce site will accept. (For testing purposes we will use Shopify).

Engineering Requirement

a. **Integration**
   - Everything will need to be integrated into existing systems, so the final product must use only existing resources available on an average android smartphone (e.g. camera, NFC sensor).

b. **Response time**
   - The application should be able to access the database and make requests within a reasonable period of time for this kind of device. The user would expect a quick response, being less than 2 seconds for most queries, with exceptions for things such as uploading photos, and searching excessively large data sets.

c. **Reliability**
   - The database should implement its own error checking and backup mechanisms to make sure that the information is not lost even in the case of hardware failure.

d. **Availability**
   - The system needs to maintain constant availability because if it is going to scale to larger business' it will need to be accessible 24/7. Acceptable thresholds for the start would be business hour availability (Mon-Fri 8-5), however, 24/7 is the goal.

e. **Out-of-the-box Usable**
   - Since this product is aimed at the less tech-savvy, it will need to be extremely user friendly and forgiving. There should be guides and easy-to access documentation for all features.

Marketing Requirements

a. **Cost**
   - The system is intended to cost nothing by building upon the existing standard of having a smartphone and at least one computer around an office. The database will aim at being able to run on simple hardware, and the phone application on any modern smartphone.

b. **Training Required**
   - By designing the system in such a way that managing inventory is as easy as making a Facebook post we are going to reduce the training required to use the system to minimal levels. Some simple concepts like typing in serial numbers of products should be as difficult as it gets.

c. **Data Mining**
   - A self-contained system will allow us to track any important information, such as the volume of product being brought in and shipped out at any given time, and the places which it is going to.

d. **Interoperability**
   - The end game is to have a system that with a few simple clicks can automagically populate an e-commerce website or other database. The system aims to support preexisting environments and cooperate with preexisting technologies to get the most out of an inventory system.

Mapping of Marketing Requirements to Engineering Requirements

Marketing Requirements	**Engineering Requirement**	**Justification**
A (Cost)	A (Integration)	By focusing on having the system work with environments that are already in use we can drive down the cost through our integration techniques.
B (Training)	A (Integration) E (Out-of-box)	The integration into existing technologies will reduce training time because users will already be familiar with how an app on a smartphone works. And driving the backend to work out of the box will minimize the skill required to manage the server.
C (Data Mining)	C (Reliability) D (Availability)	The ability to track and mine the database for important information will be most impactful to the reliability of information and how available it is due to the backend also manipulating data autonomously.
D (Interoperability)	B (Response)	The device response time will be most heavily dependent on how reliant the system is on external sources. If we are constantly updating the inventory database based upon the e-commerce site the application will be forced to wait for a response from that site as well as the server.

Engineering and Marketing Requirements Trade-off Chart

+ = Improving one improves the other - = Improving one negatively impacts the other x = Do not impact each other		a) Integration	b) Response Time	c) Reliability	d) Availability	e) Out-Of-Box Usability
		+	-	+	+	+
a) Cost	-	+	-	-	-	x
b) Training	-	+	x	x	x	+
c) Data Mining	+	+	-	x	x	+
d) Interoperability	+	+	-	-	-	+

Trade-off Chart between Engineering Requirements

		a) Integration	b) Response Time	c) Reliability	d) Availability	e) Out-Of-Box Usability
		+	-	+	+	+
a) Integration	+		+	+	+	-
b) Response Time	-			+	+	x
c) Reliability	+				+	x
d) Availabiltiy	+					x
e) Out-Of-Box Usability	+					

Competitive Benchmarks

	Manual Tracking	UPC Codes	Our Design
Unique Identifiers	No	Yes	Yes
Automated	No	Yes	Yes
Searchable	No	Yes	Yes
Works for anything	Yes	No	Yes

Various applicable constraints and standards

a. GNU Public License [2]
    - The GNU General Public License provides a generic licensing agreement for software that is developed and will be freely shared. We plan on developing our project on a public Github repo and maintaining it with GPLv3 standards
b. Google Material Design Principles [3]
    - In order to have a clean and seamless app we will be abiding by the standards set forth by Google in their material design principles. These specifications give a solid foundation for developers to create consistent and efficient applications.
c. Google Play Store Developer Policy [4]
    - For us to be able to successfully publish our product to the Google Play store we will have to follow the policies set forth by Google which guide which apps are allowed to be in the store.

References

[1] Bizslate, "Home Page," [Online]. Available: http://bizslate.com/. [Accessed 10 November 2016].

[2] Free Software Foundation, "The GNU General Public License," Free Software Foundation, 29 June 2007. [Online]. Available: https://www.gnu.org/licenses/gpl-3.0.en.html. [Accessed 20 November 2016].

[3] Google, "Introduction - Material Design," Google. [Online]. [Accessed 20 November 2016].

[4] Google, "Developer Policy Center," Google, 2 July 2016. [Online]. Available: https://play.google.com/about/developer-content-policy/. [Accessed 20 November 2016].