**Inventory System for Minor and Major Equipment**

**August Davis**

**Student**

**Department of Computer science**

[***August.Davis@und.edu***](mailto:August.Davis@und.edu)

**David Erickson**

**Student**

**Department of Computer science**

[**david.p.erickson@ndus.edu**](mailto:david.p.erickson@ndus.edu)

**Christian Hansen**

**Student**

**Department of Computer science**

[**christian.hansen@ndus.edu**](mailto:christian.hansen@ndus.edu)

***Abstract:*** *Currently the University of North Dakota Computer Science department (UND CSCI) tracks its equipment via an Excel spread sheet. Under the current system each piece of equipment is given a barcode representing a unique identification number which then then must be manually added to the inventory spread sheet. This is a time-consuming task combined with the access issue presented with storing this information in a single spread sheet file makes the current system inefficient and to an extent risky to use. This project is aimed at replacing the current system with one that will allow multiple users appropriate access to the inventory data, as well as speed up the entry process by scanning each items barcode with a phone camera. To accomplish this, a web application was built to provide an interface for the user to access the different functions. From this application, they will be able to create, delete, view, update, and locate inventory given the correct permissions.*

**INTRODUCTION:** The departments at the University of North Dakota (UND) have two broad categories of inventory that must be kept track of: Major inventory, which consists of inventory costing over $5,000, and Minor inventory, consisting of inventory under $5,000. The Computer Science department currently handles tracking its inventory by giving each item a unique identification number and manually adding it with all the relevant information about the item to a spreadsheet in an Excel spreadsheet. Some of the other information currently includes the room number, serial number, description, quantity of the same item, date purchased, original costs, and replacement costs.

The lack of user access control and inventory tracking accuracy creates a security and logistics issue. With the current system, there is one central copy that can be copied to others, but only the original will be used to track any changes. If the spreadsheet should become corrupted, then the department could lose inventory records if they are not backed up.

The goal of this project is to create a similar centralized database where multiple users can securely access inventory information and update the database in a controlled method. The user will also be able to utilize the barcodes that already exist on current inventory to locate and update inventory in the database. This will improve the speed and accuracy of inventory management.

**RELATED WORKS:** rip the good ones from the paper we did in semester one

**APPROACH:** The team started working on this project in visual studios because of its .NET services to build a server-based web application. Visual Studio also allows testing with different web servers. Visual Studio also allows easy integration of different packages such as a Scanner API. It also provides in depth debugging and controlled testing.

The previous year’s team had built a series of applications which had worked, but due to the lack of documentation and segmented source code, it was not possible to continue work on their project. So, this year’s team decide to rebuild the system from scratch and document the source code to provide an easy platform for future developers to maintain and update the system.

The first attempt at a solution was based around the idea of building a separate application for Android, iOS, and Web-based platforms. This introduced a development challenge in finding and building something that would work well between all platforms. The team eventually decided to pivot directions and focus on one web-based application that would work across all platforms and feed off the same code base. This was decided when it was realized that how we wanted to handle the barcode scanning would be very difficult to maintain across the different platforms. The team is also taking extra steps to make sure that the future developers of the system will be able to work on the project with minimal friction.

**IMPLEMENTATION:** explain how we broke the project down in to three parts.

How did we set up the website part of the project?

How did we set up the database portion of the project?

How did we set up the scanner portion of the project?

When this project was going to be a stand-alone app the plan was to integrate the scanner function into the app. When the focused pivoted to a website this became impractical do to how the asp.net framework processes information server side and being priced out of commercial solutions. The solution to this problem was to rely on external hardware and apps to handle the scanning. For pc this means a handheld USB barcode scanner to scan the code in to the appropriate text box. For android the scan button calls an external app on the phone to achieve the same affect, an IOS version of this should be possible but is not one of the goals of the project currently.

What would someone need to know to run our project?

**RESULTS:** Do the web pages work on mobile? On desktop?

Does the database receive, return, and store what it needs to?

Can barcodes be scanned under reasonable conditions and in a resemble period (i.e. less than two seconds turnaround time)?

Is there enough documentation for others to continue working on this.

**FUTURE WORK:** There will be some work to do on the system after this year. Some things that still need to be addressed is where and how to host the system. The team was in contact with professor Nordlie to see about hosting the app on UND’s servers, but we were not allowed. It was discussed with professor Grant that we might try to host it on a machine in the CSCI department. This may be best short term, but it would be preferred to move to a larger server as we scale up in inventory handled. Some extra features that could be added in the future is tracking the status of the inventory and adding an option to turn in a maintenance request if it breaks.

Features could be stream lined or added?

**CONCLUSION:** Acknowledge success or failure.

Highlight how we set this up to be built upon.