THESIS

MOBILE SUPPORT FOR LOGISTICS

by

Paxton L. Miller

March 2016

Thesis Advisor:	Arijit Das
Co-Advisor:	Gurminder Singh

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704–0188*

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** March 2016	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE MOBILE SUPPORT FOR LOGISTICS	**5. FUNDING NUMBERS**
6. AUTHOR Paxton L. Miller	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

With the responsibility of providing logistics support as quickly, efficiently, and effectively as possible to the warfighter, the Department of Defense must continually examine all feasible options to improve processes and eliminate shortfalls. As the availability and opportunity to refine supply-chain management methods through the use of mobile devices increases, a solution that utilizes them needs to be implemented.

This research reviews current military enterprise-resource-planning software systems and their functions, and details why the lack of a mobile solution is limiting their performance and capabilities. It does so by exploring how administrative logistics functions can be dramatically enhanced through the use of a mobile application designed for the logistics program Global Combat Support System – Marine Corps. The prototype also demonstrates and evaluates how mobile devices could provide benefits beyond their inherent portability advantages.

14. SUBJECT TERMS GCSS-MC, mobile application, enterprise resource planning	**15. NUMBER OF PAGES** 87
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540–01-280-5500 Standard Form 298 (Rev. 2–89) Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

MOBILE SUPPORT FOR LOGISTICS

Paxton L. Miller
Captain, United States Marine Corps
B.B.A., Texas A&M University, 2007

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
March 2016

Approved by: Arijit Das
Thesis Advisor

Gurminder Singh, Ph.D.
Co-Advisor

Peter Denning, Ph.D.
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

With the responsibility of providing logistics support as quickly, efficiently, and effectively as possible to the warfighter, the Department of Defense must continually examine all feasible options to improve processes and eliminate shortfalls. As the availability and opportunity to refine supply-chain management methods through the use of mobile devices increases, a solution that utilizes them needs to be implemented.

This research reviews current military enterprise-resource-planning software systems and their functions, and details why the lack of a mobile solution is limiting their performance and capabilities. It does so by exploring how administrative logistics functions can be dramatically enhanced through the use of a mobile application designed for the logistics program Global Combat Support System – Marine Corps. The prototype also demonstrates and evaluates how mobile devices could provide benefits beyond their inherent portability advantages.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

APB	Acquisition Program Baseline
API	Application Program Interface
ATLASS	Asset Tracking Logistics and Supply System
BYOD	Bring Your Own Device
CMD	Commercial Mobile Devices
CMR	Consolidated Memorandum Receipt
CO	Commanding Officer
COTS	Commercial off the Shelf
CSS	Combat Service Support
DOD	Department of Defense
ERO	Equipment Repair Order
ERP	Enterprise Resource Planning
FDC	Full Deployment Capability
FDD	Full Deployment Decision
FOC	Fully Operational Capable
GCSS-MC	Global Combat Support System – Marine Corps
GPS	Global Positioning System
HCI	Human Computer Interaction
IPSEC	Internet Protocol Security
MAF	Mobile Application Framework
MAGTF	Marine Air-Ground Task Force
MANET	Mobile Ad-Hoc Networks
MFS	Mobile Field Service
MIMMS	Marine Corps Integrated Maintenance Management System
NSN	National Stock Number
PC-MIMMS	PC Marine Corps Integrated Maintenance Management System
QR	Quick Response
RDBMA	Relationship Database Management System
RFI	Request For Investigation
RFID	Radio-Frequency Identification

RO	Responsible Officer
SASSY	Supported Activities Supply System
TAMCN	Table of Authorized Equipment Control Number
TLS	Transport Layer Security
VPN	Virtual Private Network

ACKNOWLEDGMENTS

I must first thank God for providing me the strength and perseverance to complete this thesis and master's degree. All the glory absolutely goes to Him in every accomplishment I make and any amount of success I have as a student and in life.

I would also like to thank my daughter, Anniston, for always providing inspiration when I needed it. She was instrumental in giving me the motivation and drive to stay committed in completing this degree. Thank you so much, sweetheart. You are the light of my life.

Lastly, I would like to thank my advisors and the Naval Postgraduate School Computer Science faculty. As a nontechnical undergraduate, the instruction and guidance I received studying computer science and conducting research for my thesis required skilled and genuinely helpful professors. I will always greatly value and appreciate their devotion to assuring the success of their students.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

As the world becomes more connected through wireless and mobile networks, the field of logistics in the Department of Defense (DOD) can benefit if it utilizes these advancing technologies. Global Combat Support System – Marine Corps (GCSS-MC) is one of the systems that can benefit from the use mobile and wireless technologies. GCSS-MC is a web-based enterprise resource planning (ERP) system used to support the Marine Corps logistics. The goal of this research is to make recommendations regarding an expansion to GCSS-MC's current Mobile Field Service (MFS) implementation plan and propose one that allows mobile devices to make real-time updates through the use of wireless networks.

The current GCSS-MC MFS implementation does not utilize wireless capabilities or mobile devices like smartphones or tablets. It does not allow changes with the enterprise servers to be made until the device returns to a connected state on a wired network, and the user manually synchronizes the updates. Technologies such as Oracle's Mobile Application Framework (MAF), which is a part of the Oracle E-Business Field Service suite, could be applied to provide an enhanced interface and framework that improves the use of GCSS-MC if mobile devices are utilized (Leach, 2006, pp. 1-1–1-3). Having the ability and mobility to make real-time updates, service requests, and ability to access current reports through wireless networks would significantly improve flexibility and utility for GCSS-MC users. In order to streamline processes in GCSS-MC and provide users with near real-time logistics information, the introduction of mobile devices is critical.

A. THESIS OBJECTIVE

The objective of this thesis is to provide program decision makers with a better understanding of the worth and best means to implement a wireless and real-time logistics capability through the use of mobile devices. It does so by developing and evaluating a proof-of-concept application that supports tasks a typical GCSS-MC user may need to perform in a mobile setting. Furthermore, the thesis exhibits the potential benefits mobile devices provide outside of their inherent mobility advantages by accessing and utilizing their sensors. Though the focus is on the GCSS-MC ERP system, much of the research and the conclusions drawn from it could apply to other DOD logistics programs.

B. METHODOLOGY

The process of this research is divided in to three phases: Research, Development, and Evaluation.

1. Research

Prior to development of the app’s prototype, it is also critical to review and understand how tasks are currently accomplished in the logistics community. This includes users that are property account holders, mechanics, supply personnel, and commanders. Background research is conducted by reviewing current ERP programs across the DOD and the Marine Corps’s current MFS implementation plan for GCSS-MC.

It is also important to review the inherent capabilities that mobile devices bring and recognize how they can be employed to improve logistics procedures as a whole. After identifying how these potential enhancements to GCSS-MC can be developed, a review of the DOD mobility strategy is completed to assure that they align with government policy.

2. Development

After researching how tasks are typically completed in the Marine Corps regarding logistics, an understanding of what actual tasks can and should be placed on the mobile application is completed from a usability perspective. Identifying different means to improve the current ways of accomplishing tasks is also reviewed. Once completed, an architecture and module layout for developing the app is finalized.

While tailoring Oracle’s Wireless Browser Based Solution into GCSS-MC could be beneficial, there are other features and functions that the Marine Corps might utilize that are not found in this option. This is most clearly seen in the property management feature of the prototype through the location snapshot webpage developed in order to enhance a user's ability to locate and manage assets through the mobile app's Global Positioning System (GPS) capabilities.

3. Evaluation

After development of the proof-of-concept mobile application, testing and evaluation is performed. These are accomplished by launching the application on an actual mobile device, completing various tasks, and validating that those tasks are properly executed on the database server. Additionally, the location snapshot webpage is tested in order to demonstrate how a mobile application could further benefit GCSS-MC overall in ways that only a mobile device feasibly can. Ideally, an operational unit would complete the testing and evaluation of the app; however, due to resource limitations, attaining this feedback falls outside the scope of this thesis.

C. THESIS OVERVIEW

This thesis is divided into five chapters. After providing a background on DOD ERP systems, Chapter II demonstrates the value of using mobile devices to complete administrative logistics tasks in a wireless environment. Chapter III begins by exploring the application specifications and requirements that a user would desire on a mobile GCSS-MC app. Following this assessment, a potential architecture and module layout is developed. Chapter III concludes by exhibiting how the app is created and why it is configured in that way. Chapter IV then evaluates the functionality and performance of the app itself as well as the location mapping webpage extension. Finally, the thesis closes in Chapter V after making brief conclusions on the completed research, recommends potential implementation extensions on the current version of the application, and suggests areas for future work regarding this topic.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

This chapter provides an acquisition history review of the ERP logistics programs in the Department of Defense (DOD) and their implementation plan. It then takes a closer look at the architecture of how the Marine Corps' ERP version, GCSS-MC, works and how it currently operates. This chapter also explores current supply and maintenance procedures in the Marine Corps and the potential efficiencies in the workplace that can be improved through mobile devices. Finally, this chapter reviews the benefits, security protections, and general DOD strategy of mobile devices and how the military currently plans to leverage their capabilities.

A. DEPARTMENT OF DEFENSE ERP SYSTEMS

All four services in the DOD have had plans to implement ERP solutions to modernize their logistics software systems. In generic terms, an ERP system is a broad set of software-integrated activities to help manage a business. These activities help manage various components of an organization such as finance, inventory, or the supply chain (Rouse). Each of these DOD programs provides capabilities unique to each service. They are all intended to streamline the way logistics decisions are made. These implementations are designed to use COTS systems to modernize the logistics management and financial tracking of those assets. These software programs are intended to also replace antiquated systems that have been in operation for over 30 years and do not provide timely, accurate information to the user (Jones, 2010, p. 2).

The DOD has had considerable delays and cost overruns in each armed service for varying reasons, which has plagued the full implementation of these software systems. In 2012, the Air Force decided to terminate the project prior to its implementation. After granting an initial $88.5 million contract to Oracle for the replacement of over 200 legacy systems, the Air Force found the program was no longer fiscally justifiable (Kanaracus, 2012). Cost overruns had increased the program cost to over $1 billion, and the Air Force estimated that "an additional $1.1B for about a quarter of the original scope to continue and fielding would not [be possible] until 2020" (Kanaracus, 2012).

The Navy's ERP solution has also faced delays and cost overruns. The cost of this SAP-based program has run more than $570 million over the originally estimate (Jones, 2010, p. 24). The original Fully Operational Capable (FOC) date was set for 2011 (Jones, 2010, p. 23). After a two-year delay, it achieved full deployment on 23 December 2013 (IT Dashboard, 2015c). GCSS-Army, which is SAP's ERP program for the Army has faced similar problems. After a FOC slip of five years (Jones, 2010, p. 18), the full deployment date is now set for a target date of September 2017 (IT Dashboard, 2015a).

GCSS-MC, the focus of this thesis, has also faced a number of its own setbacks. The full employment of the Marine Corps program, GCSS-MC Increment 1, has been delayed by over six years and had cost overruns of $1 billion since the original APB estimate (GAO, 2014, p. 74). While the anticipated full deployment date has yet to be reached, it successfully achieved its Full Deployment Decision (FDD) in March 2015 (IT Dashboard, 2015b). Full Deployable Capability (FDC), however, has yet to be reached (IT Dashboard, 2015b). Once FDC is reached, GCSS-MC will be able to be employed and implemented in austere environments with limited to no network availability (USMC Concepts and Programs).

With all the setbacks the DOD has encountered with each of these ERP solutions, they will still advance the capabilities and efficiencies of the supply chain for each service once fully implemented. However, there are still ways to further advance the implementation plans of these solutions, including the use of mobile devices. As mobile device capabilities improve, their use in the workplace is becoming more and more prevalent; yet in the field of logistics, there are few ways that service members can use these capabilities.

B. WHAT IS GCSS-MC?

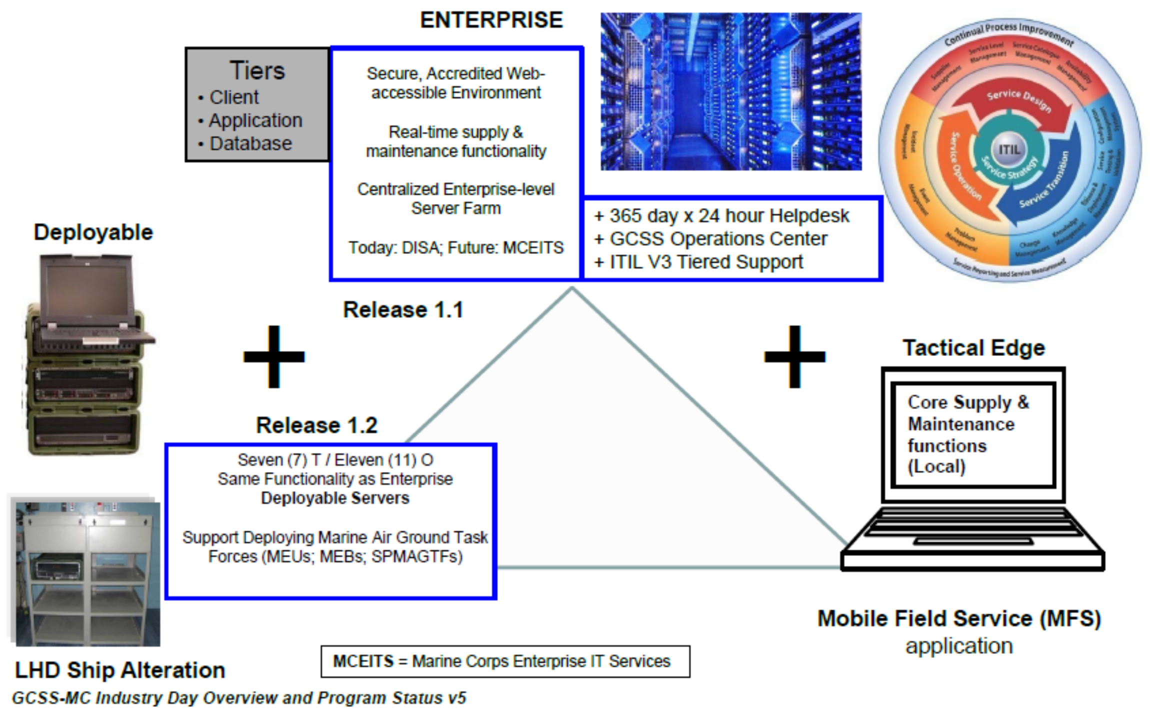
GCSS-MC is the Marine Corps logistics modernization centerpiece that provides integrated logistics information with enterprise-wide visibility in order to allow Marines to make informed decisions about the supply chain. This program uses Oracle's e-

Business Suite 11i, which is a COTS system that has been customized to meet the mission requirements of the Marine Corps (Leonard, 2014, p. 8). GCSS-MC uses proven relational database technology and applies it to logistics processes for the Marine Corps (Bitto, 2014, p. 6). The database resides on centrally located servers on a hosting site. From there, clients can access GCSS-MC from a desktop or laptop computer through the Internet (USMC Concepts and Programs). Its critical performance objectives include "reduced logistics response and customer wait time while decreasing dependence on forward-positioned stocks." (PEOEIS, 2015). This ultimately improves the combat effectiveness of the Marine Air-Ground Task Force (MAGTF).

GCSS-MC Increment 1's release is active across the spectrum of environments from garrison to deployed tactical environments. A supported unit can '"request and track the status of products (e.g., supplies, personnel.) and services (e.g., maintenance, engineering) via an Internet-based interface." (PEOEIS, 2015) The supporting unit can then process those requests, track the status of inventory or maintenance from other units or vendors, and conduct Combat Service Support (CSS) mission planning functions (USMC Concepts and Programs). Having this capability allows the user to have improved command and control support for the unit (PEOEIS, 2105).

These capabilities are not all-inclusive, but provide an understanding of how the support process works through the system. Implementation of GCSS-MC Increment 1 has been broken into two releases, and the overall architecture of GCSS-MC is found in Figure 1 (Bitto, 2014, p. 7). The first release, Release 1.1, provides enhanced supply and maintenance capabilities and reached full operational capability (FOC) on 3 June 2011. Release 1.2 provides functionality for Marines working in a deployed austere environment (PEOEIS, 2015). While most of the Release 1.2 features are currently being implemented, full deployment is anticipated in December 2015 (IT Dashboard, 2015b).

Figure 1. GCSS-MC Release Architecture. Source: Bitto, 2014, p. 7.


ENTERPRISE
Tiers
• Client
• Application
• Database
Secure, Accredited Web-accessible Environment
Real-time supply & maintenance functionality
Centralized Enterprise-level Server Farm
Today: DISA; Future: MCEITS
+ 365 day x 24 hour Helpdesk
+ GCSS Operations Center
+ ITIL V3 Tiered Support
ITIL
Service Design
Service Strategy
Service Transition
Service Operation
Continual Process Improvement
Deployable
Release 1.1
Release 1.2
Tactical Edge
Core Supply & Maintenance functions (Local)
Seven (7) T / Eleven (11) O
Same Functionality as Enterprise
Deployable Servers
Support Deploying Marine Air Ground Task Forces (MEUs; MEBs; SPMAGTFs)
Mobile Field Service (MFS) application
LHD Ship Alteration
MCEITS = Marine Corps Enterprise IT Services
GCSS-MC Industry Day Overview and Program Status v5

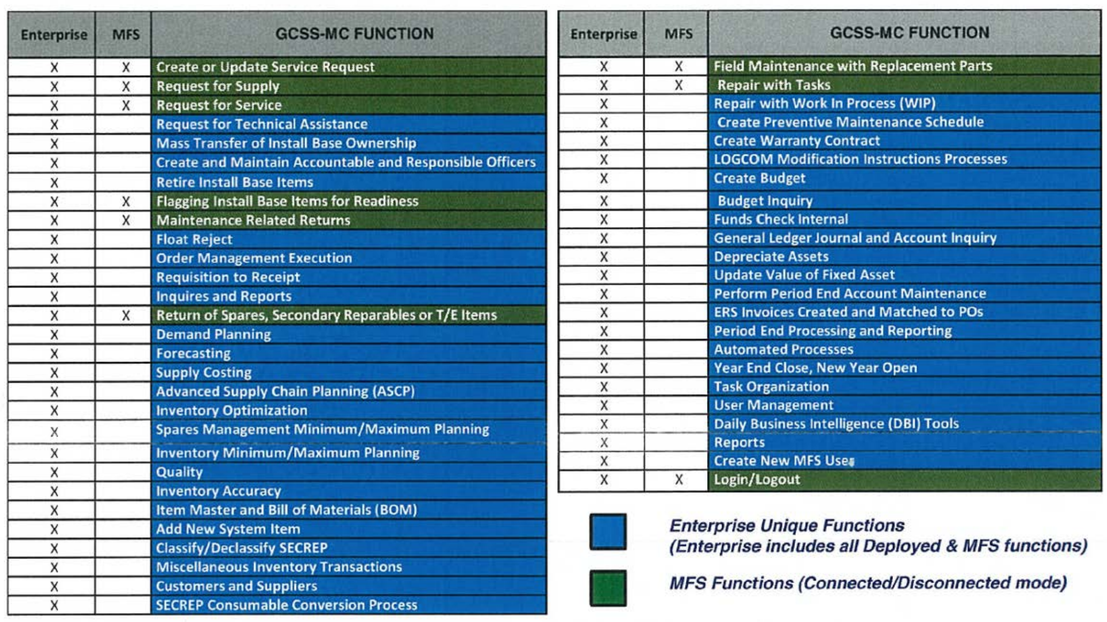
As mentioned in Chapter I, modernizing the Marine Corps logistics software with GCSS-MC is replacing many legacy systems. The largest of which are Asset Tracking Logistics and Supply System (ATLASS), Supported Activities, Supply System (SASSY), Marine Corps Integrated Maintenance Management System (MIMMS), and Personal Computer–MIMMS (PC-MIMMS). (USMC Concepts and Programs). These programs were the primary means of managing maintenance and supply readiness. Not being able to provide real-time information was a large factor in requiring a replacement for these legacy systems (Leonard, 2014, p. 7).

With the upcoming completion of GCSS-MC Increment 1, the Marine Corps will be able to utilize capabilities that allow the software to be used in two modes: connected and disconnected. In the connected mode, the user can operate GCSS via the Internet. In the disconnected mode, the Mobile Field Service (MFS) can store requests on the physical device and send them to the GCSS-MC servers once an Internet connection can be restored (USMC Concepts and Programs).

D. MOBILE FIELD SERVICE

To improve the logistics capabilities of Marines in austere locations, the MFS application has been implemented. This feature allows maintenance Marines to continue to use a limited functionality of GCSS-MC while not being connected to the Internet (GCSS-MC/LCM Increment 1 CONOPS, 2014, p. 8). The system functions that can be employed with MFS are detailed in Figure 2. One of the largest features of MFS is the store-and-forward capability. This allows users to complete forms and requests and save them until network connectivity is reestablished. This improves efficiencies drastically for users that are in austere network environments. It also improves the usability for users in garrison. Often, in a maintenance bay or on a ship, the ability to access a connected GCSS-MC laptop is unfeasible. Prior to the MFS implementation, mechanics or supply Marines would need to write down on paper the requests or updates that were needed. MFS also has the ability to transfer data from a laptop using a memory storage device such as a CD or external hard drive (Mobile Field Service-Enhanced). This allows changes to be made from a disconnected device and transfer those requests to a connected GCSS-MC machine as needed.

Figure 2. MFS GCSS-MC Functions. Source: GCSS-MC/LCM Increment 1 CONOPS, 2014, p. 51.



Enterprise	MFS	GCSS-MC FUNCTION
X	X	Create or Update Service Request
X	X	Request for Supply
X	X	Request for Service
X		Request for Technical Assistance
X		Mass Transfer of Install Base Ownership
X		Create and Maintain Accountable and Responsible Officers
X		Retire Install Base Items
X	X	Flagging Install Base Items for Readiness
X	X	Maintenance Related Returns
X		Float Reject
X		Order Management Execution
X		Requisition to Receipt
X		Inquires and Reports
X	X	Return of Spares, Secondary Reparables or T/E Items
X		Demand Planning
X		Forecasting
X		Supply Costing
X		Advanced Supply Chain Planning (ASCP)
X		Inventory Optimization
X		Spares Management Minimum/Maximum Planning
X		Inventory Minimum/Maximum Planning
X		Quality
X		Inventory Accuracy
X		Item Master and Bill of Materials (BOM)
X		Add New System Item
X		Classify/Declassify SECREP
X		Miscellaneous Inventory Transactions
X		Customers and Suppliers
X		SECREP Consumable Conversion Process

Enterprise	MFS	GCSS-MC FUNCTION
X	X	Field Maintenance with Replacement Parts
X	X	Repair with Tasks
X		Repair with Work In Process (WIP)
X		Create Preventive Maintenance Schedule
X		Create Warranty Contract
X		LOGCOM Modification Instructions Processes
X		Create Budget
X		Budget Inquiry
X		Funds Check Internal
X		General Ledger Journal and Account Inquiry
X		Depreciate Assets
X		Update Value of Fixed Asset
X		Perform Period End Account Maintenance
X		ERS Invoices Created and Matched to POs
X		Period End Processing and Reporting
X		Automated Processes
X		Year End Close, New Year Open
X		Task Organization
X		User Management
X		Daily Business Intelligence (DBI) Tools
X		Reports
X		Create New MFS User
X	X	Login/Logout

🟦 *Enterprise Unique Functions (Enterprise includes all Deployed & MFS functions)*

🟩 *MFS Functions (Connected/Disconnected mode)*

While the capabilities the Marine Corps MFS provides are improvements that the logistics field has needed, there are still areas that can be further advanced. Many of the features of MFS have come from the COTS version of Oracles' Mobile Field Service. This is a mobile solution with features that include "spare parts ordering, sourcing, receiving, returns, and knowledge management access for problem diagnosis" (Oracle Mobile Field Service Data Sheet). The largest difference between the Oracle and Marine version of MFS is the ability to utilize mobile devices and wireless networks. Having the ability to use wireless networks, Oracles' version is able to use Windows mobile devices, smartphones and tablets in addition to laptops. By only authorizing laptops to be used in the Marine Corps MFS, GCSS-MC is being restrained and not being employed to the full capabilities of current technology.

E. INVENTORY MANAGEMENT

In order to maintain accountability of equipment in the Marine Corps, a Consolidated Memorandum Receipt (CMR) is created that provides units an itemized listing of all assets maintained in ATLASS that are under the charge of the Responsible Officer (RO). The Commanding Officer (CO) of a unit typically appoints each RO. Some of the responsibilities include receiving for and maintaining accountability of all equipment in the account, conducting account reconciliations and periodic inventories, maintaining equipment, and reporting changes to the Supply Officer (Supply CMRs and ROs).

Most commands require a physical sight inventory of all assets every 90 days. Based on the size of these accounts this can take up to five workdays to simply complete the walkthrough. Following this supply record reconciliation, all discrepancies, such as correcting an incorrect serial number or adjusting the quantity on hand of an item, are formally noted in a Request For Investigation (RFI) in order to adjust the record. Depending on unit policy, the RO will then draft a document to the Commanding Officer via the unit's Supply Officer, so these changes can be approved. This process can take over 30 days to complete.

During this inventory, a RO will usually walk through the workspaces with a hardcopy listing of the items that are to be accounted for. With this process, any updates or changes following this hardcopy printout, such as equipment transfers, will clearly not be reflected. For CMRs that have a large amount of turnover, this can create issues where the Responsible Officer is reconciling an outdated account. Having a real time capability to update and inventory these accounts would prevent these concerns.

Both the Marine Corps and Oracle version of the MFS applications do not feature an inventory management module. These solutions primarily focus on the supply, maintenance, and transportation functions of logistics (e.g., parts ordering, maintenance status, or current location in the supply chain). A module that concentrated solely on improving the inventory management would be invaluable to any RO or CMR account holders.

F. PROPERTY AND MAINTENANCE MANAGEMENT

There are six functions of logistics that the Marine Corps recognize. Those are: Supply, Maintenance, General Engineering, Transportation, Services, and Health Services. GCSS-MC's focuses are the Supply and Maintenance functions.

An implied task for GCSS-MC in fulfilling the supply function is to manage property and assets. This is completed by facilitating tasks such as ordering and disposing of parts, tracking distribution and shipments, and assisting overall in determining supply requirements. The other major logistics function that GCSS-MC features is maintenance. GCSS-MC currently allows the user to perform tasks such as making equipment repair orders online, checking the maintenance status of an asset, and reviewing the overall maintenance readiness of a unit.

GCSS-MC has greatly assisted in streamlining and expediting Equipment Repair Orders (EROs). Prior to GCSS-MC, time-consuming paper submissions for EROs were required. Now, this process uses automated requests and requisitions to complete these maintenance repair orders (PEOEIS, 2015). GCSS-MC has also automated preventative maintenance scheduling. This saves time and eliminates potential human errors, thereby extending equipment service life and improving overall operational readiness. Another feature of GCSS-MC that improves supply chain management is the near real-time reports and views capability. By replacing daily and weekly reporting cycles that could potentially be outdated, GCSS-MC provides quick access to reports, which provides improvements to overall visibility and the current status of shipments or readiness. These visibility improvements of the supply chain drastically increase the ability of the commanding officer and logisticians to make informed decisions about the assets in the unit (PEOEIS, 2015).

G. SMARTPHONE SENSORS

If mobile devices could be used with GCSS-MC, the sensors that are inherent to any smartphone could advance the logistics capabilities of the Marine Corps tremendously. For example, most laptops do not have a Global Positioning System (GPS) sensor, which could have a number of benefits to the GCSS-MC program. Almost all smartphones, however, do have a GPS capability. While laptops have far fewer sensors than mobile devices, the general design and purpose of laptops is an additional limit to potential GCSS-MC improvements. These designs can prevent those laptop sensors from being used in a meaningful way. A smartphone camera, for example, can easily take pictures of almost anything a user chooses. A camera on a laptop, however, is designed primarily for videoconferences, which limit its capabilities.

A smartphone camera feature for GCSS-MC would provide enormous advantages. For the function of inventory management, having a camera could allow a CMR Responsible Officer to better reconcile and identify account discrepancies. Typical situations where this would be useful are validating where serial numbers are incorrectly entered or showing that an asset is a different model than the one described on the CMR. Additionally, being able to take a picture of the physical item, which could be uploaded to a server, would provide the account holder a better way to identify and recognize the assets that are the user’s responsibility. Often the description of an item can be vague or broad. For example, a Marine may be responsible for a general-purpose tent. While the Table of Authorized Equipment Control Number (TAMCN) or National Stock Number (NSN) associated with that tent can allow the user to look up the specifics for that asset, having an actual picture of the item would be more beneficial. One reason would be to capture the sub-items that belong to that tent, such as stakes, hammers, or simply the box that contains it. These sub-items are components lists or “SL-3 gear,” which is collateral material associated with a principal end item (Mutter, 1996, pp. 2–7). This is important, because a CMR account holder may only be responsible for maintaining the end item. However, this equipment is critical and often overlooked when reconciling the items a unit is responsible for because this collateral material is not itemized on a CMR account. Having the capability to take pictures of all the items, including the “SL-3 gear,” associated with any asset across the Marine Corps would greatly improve the operational capability of a unit as well as save a tremendous amount of funds that are used to replace lost parts.

Overall, having a picture capability for GCSS-MC would assist the account holder in other areas that could otherwise be problematic. A photo feature would answer questions such as “How many stakes should belong with this tent?,” “What color is the tent?, and “What does the box that contains it look like?” These answers are not found in GCSS-MC and rely upon the knowledge of the account holder.

Another way a camera can be used to improve GCSS-MC would also require the Marine Corps to adopt a new means to read bar codes. If Marine Corps assets had Quick Response (QR) codes, then any GCSS-MC user with a smartphone could pull those items details. These details, for instance, could include the NSN, description, serial number, unit price, etc. Doing this would save users a large amount of time and would assist in improving the overall asset control of property throughout the Marine Corps.

The GPS of a phone is another sensor that could be exploited to improve GCSS-MC. Because mobile devices are not currently used for GCSS-MC, the advantages that a GPS could bring to the ERP system are not used. Oracles’ version of MFS however can use the phone’s GPS for certain features such as driving navigation. This would assist drivers in delivering parts to a destination. An area that this version does not use GPS is for property or inventory management. Though the Marine Corps uses Radio Frequency Identification (RFID) tags and other means to track items in transit, many of the assets that a unit maintains, especially in garrison, will remain in the same place for an extended period of time. Since the vast majority of assets will have RFID tags only temporarily, if at all, a means to capture an item’s location via a phone’s GPS would be beneficial. Of course, the location capture via a phone’s GPS would be as up-to-date as the last time it was entered. So this would not provide a real-time solution, but it would still have significant value. With an option to capture an item’s location that will not relocate often, such as an office computer or printer, the visibility and accountability of all property can be improved greatly.

H. DEPARTMENT OF DEFENSE MOBILE DEVICE STRATEGY

The Department of Defense Mobile Device Strategy is identified as “information technology goals and objectives to capitalize on the full potential of mobile devices.” according to Teresa M. Takai, the DOD Chief Information Officer. She also states that this strategy is “about keeping the DOD workforce relevant in an era when information and cyberspace play a critical role in mission success” (Takai, 2012, p. i).

The first goal of this strategy is to advance and evolve the DOD Information Enterprise Infrastructure to Support Mobile Devices (Takai, 2012, p. 2). The objectives needed in order to meet this goal are to: evolve spectrum management, expand infrastructure to support wireless capabilities, and establish a mobile device security architecture (Takai, 2012, p. 2). By expanding infrastructure to support wireless capabilities, such as Wi-Fi or LTE-based 4G cellular infrastructures, into the DOD’s enterprise software, GCSS-MC could be a large benefactor of that. In tactical environments architectures such as mobile ad-hoc networks (MANETs) could be explored.

This strategy shows that the DOD aims to use mobile devices in the workforce, and it is the DOD Mobile Device Implementation Plan executes the goals of this strategy. Currently, there are means in each service to provide commercial mobile devices (CMDs) to promote the use of mobile non-tactical applications in the Department of Defense (Takai, 2013, p. 1). Each service uses various programs to fulfill this goal. For example, the United States Marine Corps uses the Trusted Handheld Mobility Pilot to provide a classified CMD capability to users. With these mobile devices, the Marine Corps is able to utilize some government applications for operational use (Takai, 2013, p. 7). One way the Marine Corps is currently evaluating the Trusted Handheld program by fielded mobile devices through ViaSat Inc. ViaSat's goal is to provide secure communications over 3G/4G/LTE cellular and Wi-Fi networks, secure storage of sensitive data, and protection against malware attacks (ViaSat). If the Marine Corps decides to field these devices throughout the service, then this provides the opportunity for GCSS-MC to be involved.

Bring Your Own Device (BYOD) is also a capability that the Department of Defense is exploring. BYOD are personally owned mobile devices that could be used for enterprise business purposes (Takai, 2013, p. 16). A service member having the ability to utilize his/her own personal mobile device for unclassified or potentially classified applications would have dramatic benefits throughout the defense industry.

Because security is a chief concern for any communication software in the Department of Defense, it is understandable why wireless networks and mobile devices are not authorized in the Marine Corps version of MFS. However, with the wide implementation of trusted handhelds and the possibilities of securing personal mobile devices, this is an opportunity that should not be missed. While the scope of this thesis does not focus on mobile security, there are capable means of protecting the confidentiality and integrity of the devices. Encryption requirements such as a TLS (Transport Layer Security) connection or a VPN (Virtual Private Network) IPSec (Internet Protocol Security) tunnel would greatly improve the authentication and overall protection of the program. GCSS-MC could also require users to only utilize DOD private wireless networks vice a public or commercial carrier network. While these measures are not all encompassing for protecting mobile devices, they are a ways that drastically improve their security. With the limited number of connected devices in the-workplace and the abundance of smartphones of each service member, utilizing mobile devices would be a force multiplier for the Marine Corps and security concerns should not prevent them from being used for GCSS-MC.

I. SUMMARY

This chapter shows that after numerous setbacks, most DOD services remain dedicated to implementing an ERP solution. Even with this modernization in software management, there are further efficiencies that can be made to current logistics processes even with this commitment to improvement. Program managers should be lobbying to fully utilize the capabilities of current technology in their respective ERP system. Chapter III will explore some of these potential enhancements by developing a proof of concept mobile application for GCSS-MC that not only provides mobility of the current system, but new features that can only be utilized on a mobile device.

III. DEVELOPMENT

Users across the DOD lack a means to utilize mobile devices on the current logistics ERP systems. The Marine Corps's current logistics system program, GCSS-MC, has implemented a mobile field service plan for disconnected devices. The term "mobile," however, in this plan can be misleading as it does not allow for the use of smartphones or tablets, or utilize wireless capabilities in general. It requires a disconnected laptop user to make "offline" updates, and modifications are not applied until re-connected to a wired network. More specifically, this plan does not apply changes with the enterprise servers until the device returns to a connected state, and the user manually synchronizes the database adjustments. While the MFS plan does improve the usability of GCSS-MC, a tailored application suite of GCSS-MC designed for mobile devices would greatly improve efficiency.

In this chapter, how a proof of concept mobile application for GCSS was developed is examined. First described are the application specifications, requirements, and architecture. Following this explanation, the design for both the application and the database it utilizes is discussed. The last item reviewed is a property management webpage that utilizes the mobile app's location feature.

A. APPLICATION SPECIFICATIONS AND REQUIREMENTS

Prior to developing the model and architecture for a custom mobile suite of GCSS-MC, there are critical questions that must be asked. A thorough task analysis is needed prior to developing an interface or architecture for the application. Assessing user needs and reviewing how tasks are currently accomplished will complete this. Two of the most important questions to ask are, "What functions and operations are best served on a mobile platform?" and "What current GCSS-MC functions should not be on the mobile app?" Current processes previously discussed lead us to the following conclusion. Having the ability and mobility to make real-time updates, maintenance requests, and access to current readiness and property reports through wireless or cellular networks would significantly improve flexibility and utility for a wide scope of users. This capability would alleviate supply and maintenance constraints significantly.

Another question to ask is, "Who would utilize this application if a mobile version of GCSS-MC is created?" The answer is straightforward. The intended users would be any personnel who have responsibilities pertaining to the handling, receipt, accountability, ordering, disposal, and maintenance of equipment. Any GCSS-MC user who needs to perform maintenance or supply requests could also utilize this system. Additionally, ROs that are performing accountability reconciliations or would like to get an up-to-date status of the assets on his or her CMR could use this mobile version of GCSS-MC. Lastly, if supply, maintenance, and readiness reports were available to be accessed, then the benefit of this application would broaden to almost any GCSS-MC user, such as supply, maintenance, and logistics Marines along with the command leadership. All of these stakeholders have a vested interest in the operational readiness of the assets in the unit.

Specific functions can be formulated that should be on the mobile application. After analyzing the answers to these previous questions, the following requirements would need to be provided for the GCSS-MC user:

- Database searches by TAMCN, NSN, or Serial Number (S/N).
- A user-friendly inventory accountability function that allows a RO to quickly view the assets under his or her responsibility. This feature should allow the user to "check," flag, or make comments to each end item within the app, which is needed in order to aid in conducting inventory reconciliations. Additional features could include an option to upload photos. This could provide a RO a better understanding of what the assets he or she is responsible for look like and improve quality assurance overall.
- Supply function that enables users the ability to request parts or change the status of an item or end-asset (e.g. Transfer, Pickup, Return to Stock).
- Maintenance function that allows users to check or update the status of an end item. A comments section could be used to provide an exact problem description of an asset. A camera feature would allow take a picture of a broken item as well.

- A user-friendly snapshot of any recent supply or maintenance updates.
- The ability to add or delete an asset from the database.

A refined GCSS-MC mobile app should not be limited to these capabilities, but they do frame the development of what is needed for a tailored mobile solution for this logistics application. Having a user-friendly version of GCSS-MC's current capabilities on a smartphone would be a force multiplier in itself. By adding smartphones or tablets into this program, there are also new capabilities that can be leveraged that only they can bring. The two most important capabilities are the utilizing the inherent portability of the devices and employing the sensors in ways discussed in Chapter II.

B. ARCHITECTURE

After reviewing the requirements for the application, a proof of concept architecture needed for the app's purpose can be constructed. The first question to consider is deciding what type of handheld device should and will be used in the GCSS-MC app. If employed, trusted handhelds would very likely be one category to be used. In an ideal scenario, however, all devices could utilize the GCSS-MC mobile app. This would include all tablets and smartphones and be compatible with all mobile operating systems such as Android, iOS, BlackBerry, and Windows Phone.

To prevent the need of programming natively for each possible device, there are bridging technologies that assist in the mobile development framework. For a mobile developer that "needs to extend an application across more than one platform, without having to re-implement it with each platform's language and tool set," Apache Cordova is an option (Cordova, 2015). While other frameworks are available, this is the open-source mobile development framework that will be used.

Cordova, formally PhoneGap, allows technologies such as HTML5, CSS3, and JavaScript to be used that can be applied for cross-platform development. Applications are executed within "wrappers targeted to each platform, and rely on standards-compliant API (Application Program Interface) bindings to access each device's sensors, data, and network status" (Cordova, 2015). For the purposes of testing this application, Apache Cordova was used for the interface programming for an iPhone.

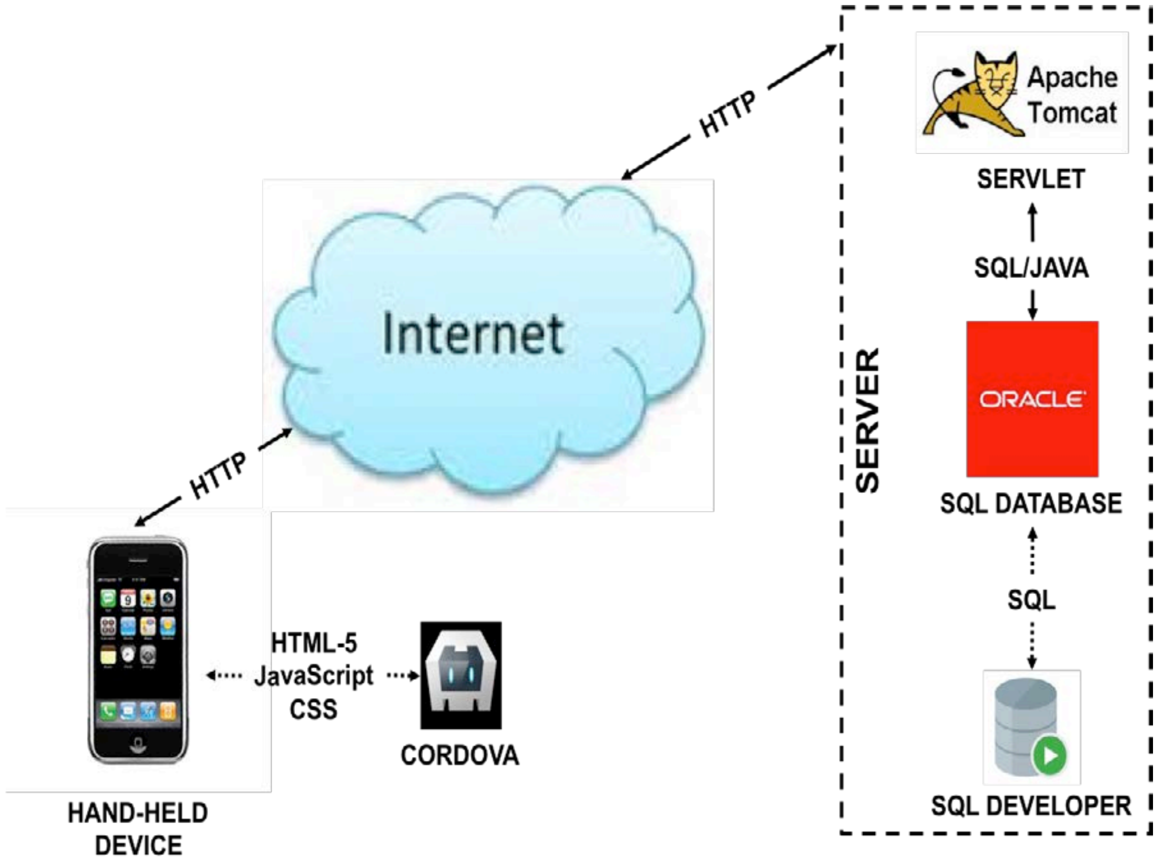
The other major consideration for the app architecture is how the handheld device will communicate with GCSS-MC. Because the actual GCSS-MC servers cannot be used for the purposes of this thesis, a Relational Database Management System (RDBMS) and a web server will also be needed to create this application.

The genuine GCSS-MC system uses an Oracle database to host its logistics information, so it was important for this project to do the same. Therefore, the Oracle 11g Database is the RDBMS that the GCSS-MC mobile application will be communicating with. Another critical requirement for the architecture is the web server. This is the component that interfaces between the Oracle Database and mobile device. Apache Tomcat is a software implementation of the Java Servlet, JavaServer Pages, Java Expression Language, and Java WebSocket technologies. (Apache Tomcat) Tomcat 7.0 was used due to it being open-source, availability, and ease of use.

In order to better communicate with the Oracle database, the SQL Developer application was also utilized. SQL Developer is an Integrated Development Environment (IDE) that provides a graphical user interface (GUI) and allows a user to complete database tasks with fewer clicks and keystrokes. Its purpose is to "help the end user save time and maximize the return on investment in the Oracle Database technology stack" (Oracle, 2014). Additionally, SQL Developer supports the database used for this project, Oracle Database 11g (Oracle, 2014). For these reasons, SQL Developer was the preferred IDE choice.

To best illustrate this architecture and how they interrelate, see Figure 2. It shows that Cordova is used as the base-programming tool for developing the application's interface. The mobile device then sends an HTTP request to the Apache Web Servlet. The Server then communicates with the Oracle Database via Standard Query Language (SQL) and returns a response to the mobile device.

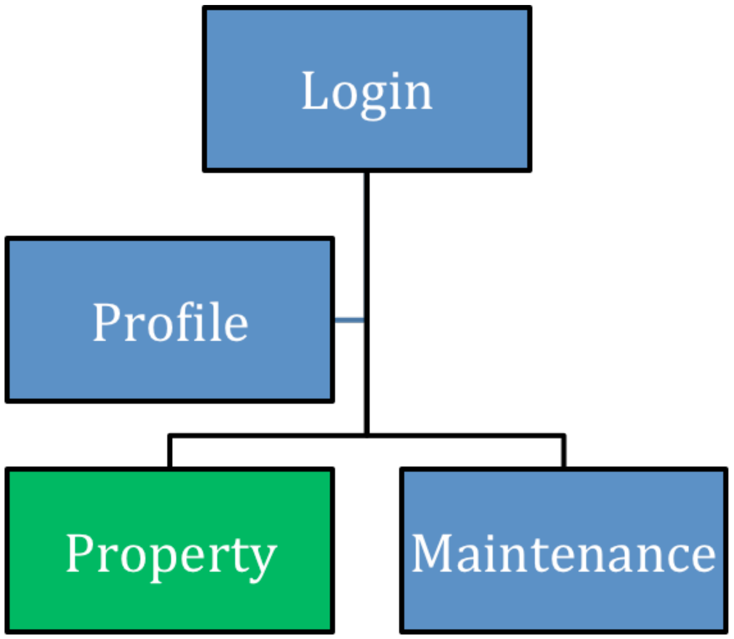
Figure 3. GCSS-MC Mobile App Architecture



C. DESIGN

The design of the GCSS-MC app must be as user-friendly as possible and will provide an ability to easily execute the commands that a typical Marine would request from a mobile device. With this in mind, three modules are needed: the Profile Management module, a Property Management module, and a Maintenance Management module, as shown in Figure 4. Highlighted in green on the figure is the Property Module, which is the focus of the back-end programming for this thesis.

Figure 4. GCSS-MC Mobile App Modules

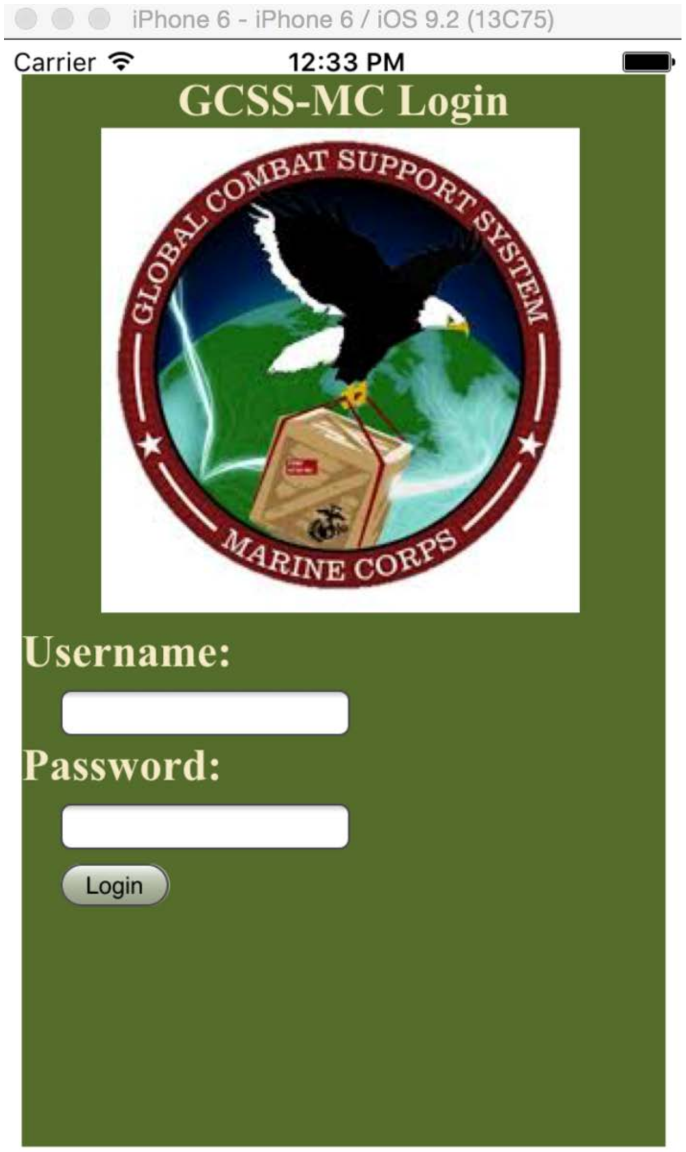


1. Login

For security and proper profile navigation, a login screen is required for the GCSS-MC mobile app. Because security is such a concern for the military, especially with tactical applications, ensuring the integrity and confidentiality of the application is crucial. As technology continues to advance, the ways a user can securely login also increases. Biometric matches that read the users face, eyes, or fingerprints are becoming more prevalent. At the current time, however, password management is the intended means for logging into the GCSS-MC mobile app.

Figure 5 shows what the user sees once the mobile app is launched. Each GCSS-MC user will need a login and password that will launch the main screen.

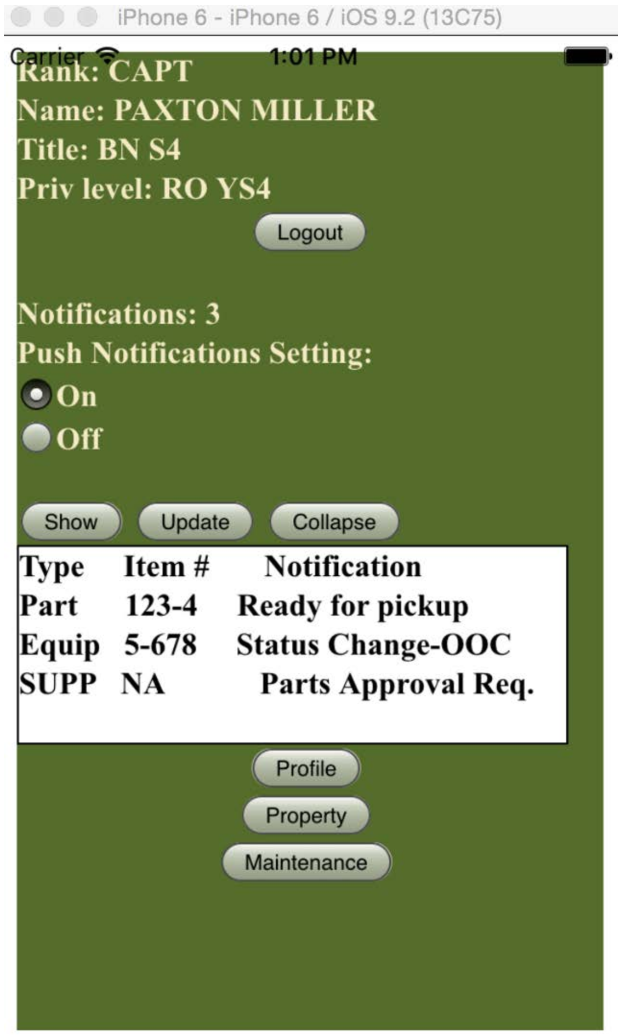
Figure 5. GCSS-MC Mobile App Login Screen


iPhone 6 - iPhone 6 / iOS 9.2 (13C75)
Carrier
12:33 PM
GCSS-MC Login
GLOBAL COMBAT SUPPORT SYSTEM
MARINE CORPS
Username:
Password:
Login

2. Main Screen

Upon logging into the app, GCSS-MC will begin by pulling information personalized to that user’s unit, privileges, and responsibility. That information will then be displayed onto the main screen as shown in Figure 6.

Figure 6. GCSS-MC App Main Screen



Rank: CAPT
Name: PAXTON MILLER
Title: BN S4
Priv level: RO YS4

Notifications: 3
Push Notifications Setting:
[✓ On]
Off

Type	Item #	Notification
Part	123-4	Ready for pickup
Equip	5-678	Status Change-OOC
SUPP	NA	Parts Approval Req.

The top of the main screen shows the user's profile information. This includes the user's rank, job title, and privilege level within GSCC-MC. Privilege level is discussed further in the Profile Management portion of the thesis. Below the profile information is the logout button and "Push Notifications Setting." This setting is intended to allow the user to be notified via an alert or banner message when an update of any changes that have occurred within GCSS-MC. This option would likely be utilized by the user that may desire to know exactly "when a needed repair part arrives for pick-up," "when an asset's readiness status is changed," or "when a part was approved for order." For the user who would like to know as soon as these types of notifications are updated in the system, having push notifications turned on would be recommended.

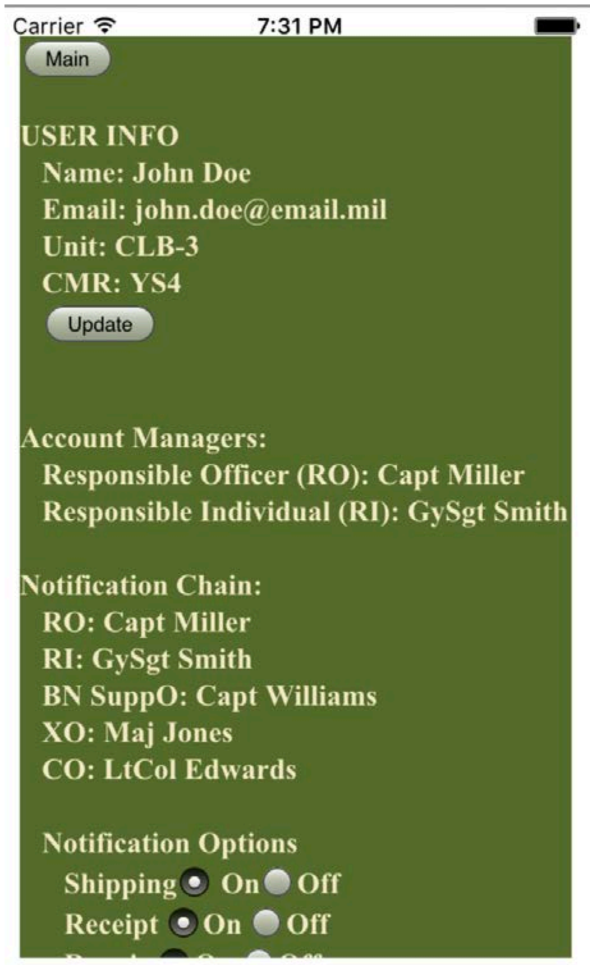
If push notifications are in fact turned on, it is important for the user to know that this may not be the ideal setting for an austere environment. If this setting is turned on, the phone will be in continual communication with the servers, taking up valuable battery power and bandwidth. Even with this setting off, the user will be able to pull data as needed from the GCSS-MC servers.

Under the push notifications settings, are the Show, Update, and Collapse buttons. The Show and Collapse buttons will display or remove the notifications screen respectively. The purpose of this is the user privacy and preference of what he or she would like to view on the main screen. The Update button will ping the servers to check for any new information. Below these buttons is the actual notifications screen, which displays the most recent updates to supply or maintenance that user is privy to knowing. Below this notification screen are the buttons that will take the user to the three main modules. These are the “Profile Management,” “Property,” and “Maintenance” components of the app.

3. Profile Management

The profile management portion of the GCSS-MC mobile app serves multiple purposes. It allows the user to view the current user profile information, the ability to update it, and option to change the types of notifications located on the main screen. Figure 7 shows the layout of this module. This screenshot depicts the page upon navigating to it from the main screen.

Figure 7. Profile Management Module



USER INFO
Name: John Doe
Email: john.doe@email.mil
Unit: CLB-3
CMR: YS4

Account Managers:
Responsible Officer (RO): Capt Miller
Responsible Individual (RI): GySgt Smith

Notification Chain:
RO: Capt Miller
RI: GySgt Smith
BN SuppO: Capt Williams
XO: Maj Jones
CO: LtCol Edwards

Notification Options
Shipping [✓ On] Off
Receipt [✓ On] Off

At the top of the module is a button that allows the user to navigate back to the main screen. For usability purposes, this button is placed at the top left portion of the page. This allows the user to quickly move throughout the application with ease. Below the button is the user's information, which includes basic details such as the user's unit, email, and CMR account. An update button is found just below in the event that there is a discrepancy. For example, a user may need to use the update button if his or her GCSS-MC account has outdated information, such as the user incorrectly listed in the wrong unit. This update would send the request to the proper personnel that are authorized to change a user's unit and permissions.

Based on the account information in the USER INFO portion of the page, specific and custom details regarding the Account Managers and Command Notification Chain will be presented. Figure 7 also shows an example of what this would look like. For the CMR account YS4 at the Unit CLB-3, all chain of command authorities for that specific account are shown. This includes the RI, RO, Battalion Supply Officer, Executive Officer (XO), and Commanding Officer (CO).

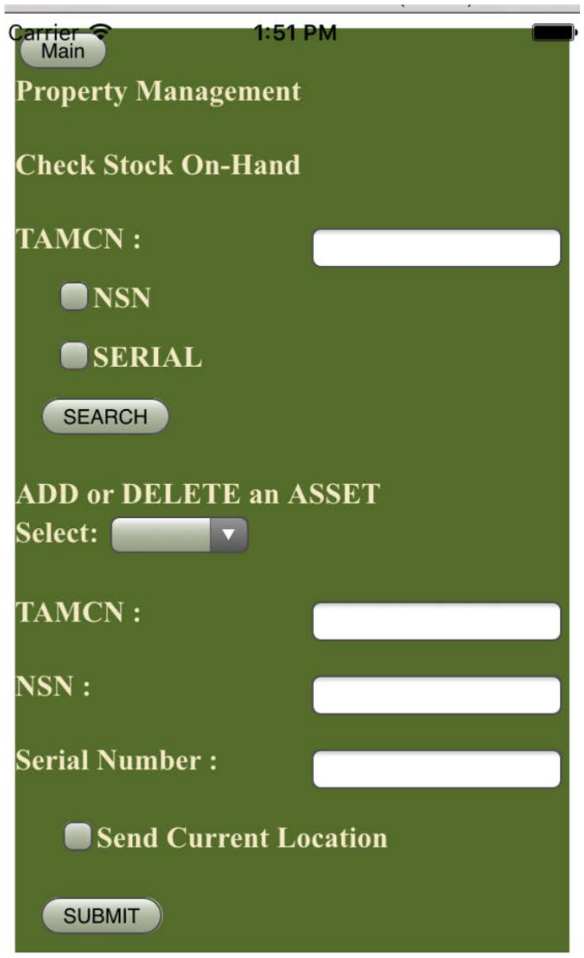
The last section of the Profile Management Module grants the user different notification options. These various notifications would be displayed in the form of banner messages on the user's phone. In this way, updates will immediately notify the user whether that person has the application presented on the screen or not. Similar to how a user may receive notifications for text or email messages on their mobile device, the GCSS-MC mobile app will be able to inform the user of updates in the system. This eliminates the need for a user to continually check the system for the updates he or she is anticipating. For example, if a part is in transit to fix a high-priority asset, a commander may want to know when it arrived at the supply or maintenance bay. Having a notification setting would grant that user or any other key individual access to that information once is was updated in the system. The current notification options are for shipping, receipt, and repair updates.

4. Property Module

For the purposes of this thesis, the Property Module is the most critical portion of the application. All research that communicated with the database server was conducted in this module, while the others are a snapshot shell of what a finished product could look like. This module also communicates with the GPS sensor of the mobile device.

Upon navigating from the main screen, the top of the property module displays a number of options as shown in Figure 8. At the top of the screen is the "Main" button, which will take the user back to the main navigation screen. Below this is the Check Stock On-Hand section.

Figure 8. Property Module, Part 1



Property Management

Check Stock On-Hand

TAMCN :

NSN

SERIAL

ADD or DELETE an ASSET

TAMCN :

NSN :

Serial Number :

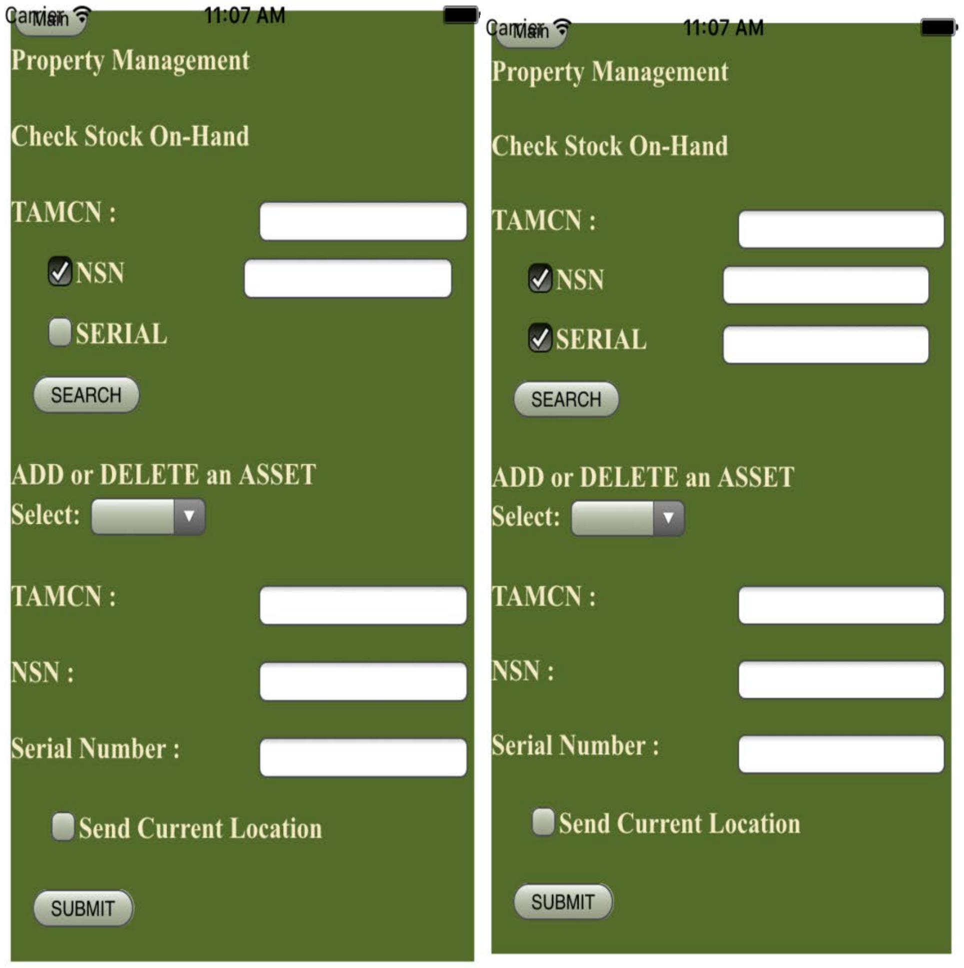
Send Current Location

In the Check Stock On-Hand portion of the application, the user is first prompted with providing the TAMCN. All principal end items in the Marine Corps are assigned a TAMCN as a supply description code. Each TAMCN starts with a letter to represent its commodity designator followed by several numbers unique to that end item. For example, the M9 Pistol’s TAMCN is E1250. Because every end-item has a TAMCN, the search requires the user to know that information in order to search for an asset.

Below the TAMCN prompt, the user can select the NSN and Serial Number check boxes. Figure 9 displays each selection respectively. The snapshot on the left displays a NSN search, and the picture on the right shows the more specific Serial Number search.

Upon selection of these boxes, a prompt for entering their associated information is displayed.

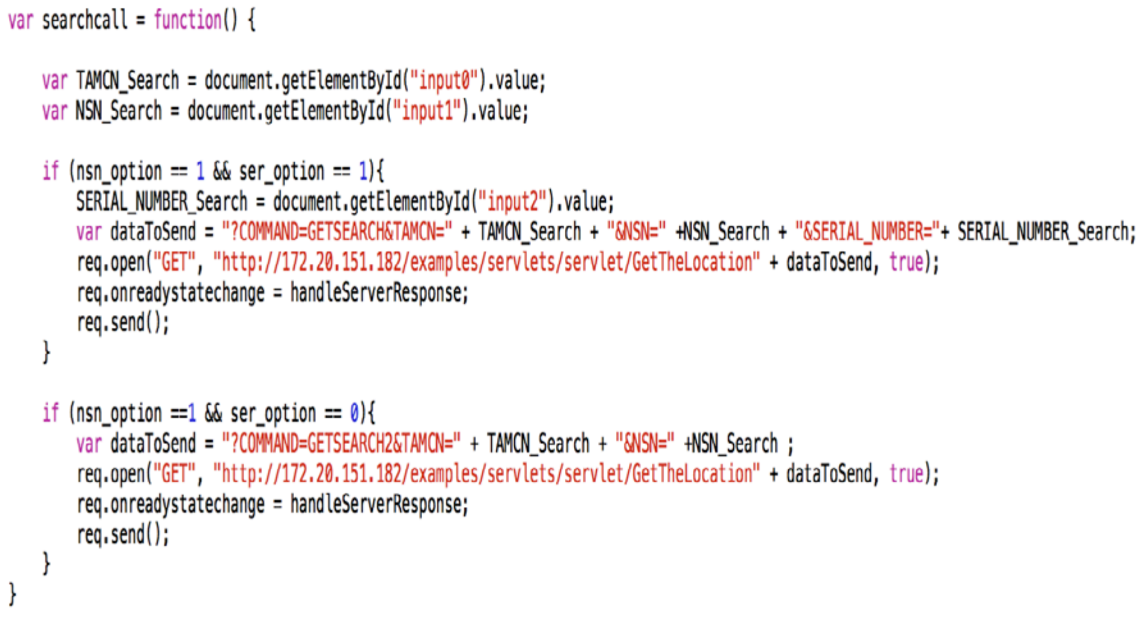
Figure 9. NSN and Serial Number Search



If the user requests a search by only entering the TAMCN and NSN, the phone will display data specific to that request. For this particular request the following information will be displayed after the user selects the search button: TAMCN, NSN, description or nomenclature of the asset requested, the quantity rated in that CMR or unit, all serial numbers belonging to that unit, the quantity the CMR or unit is overall responsible for, and the quantity on hand. When a user would like to make a more detailed search on a specific serial number, a different result will be displayed on the phone. In this case, the following information will be displayed: TAMCN, NSN, Serial Number, description, Maintenance Status, "on hand" status, location of the asset, and time stamp of the most recent update. The location of the asset is not a real time capability, but where it shows where it was located on the date of the time stamp.

The first step in completing either type of search for the user is to properly formulate the request. The phone will send a HTTP GET request to the IP address and servlet where the database is found. Figure 10 displays a function call, which packages each of these HTTP requests for the searches in a way the server can understand. A different command will be sent to the server depending on whether the user is requesting a general NSN search or a more specific Serial Number search.

Figure 10. Search Call Function

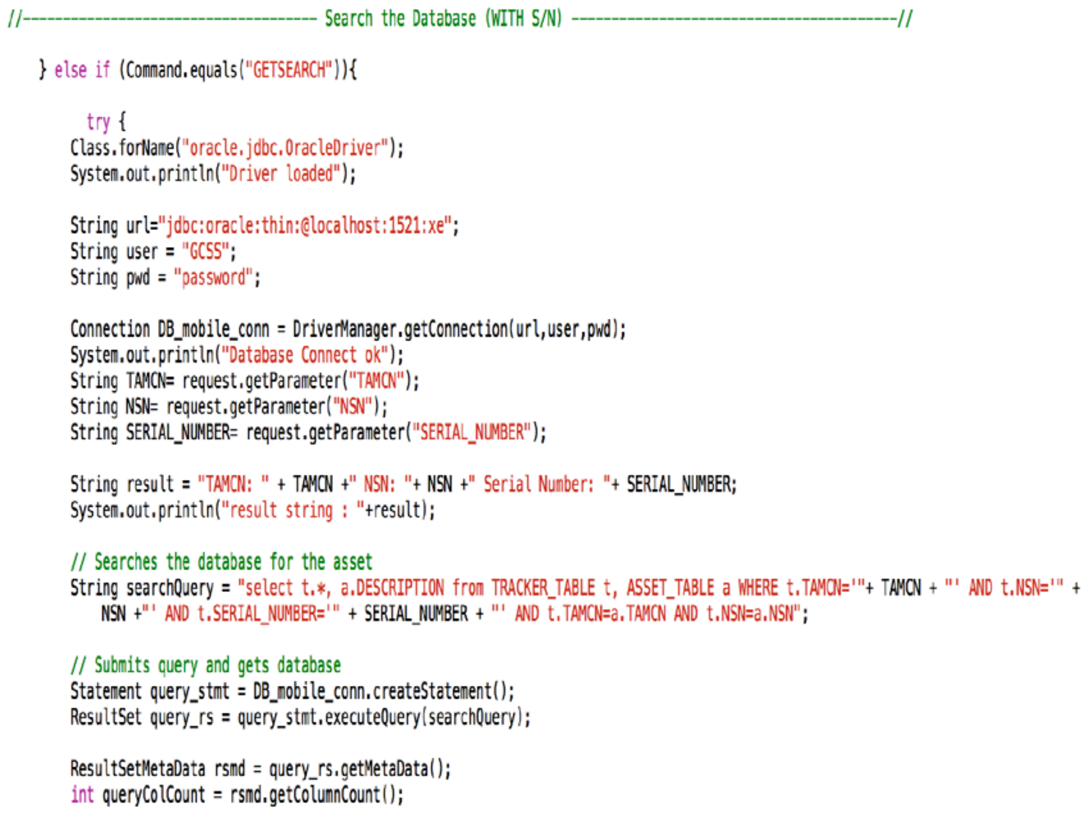


```
var searchcall = function() {

    var TAMCN_Search = document.getElementById("input0").value;
    var NSN_Search = document.getElementById("input1").value;

    if (nsn_option == 1 && ser_option == 1){
        SERIAL_NUMBER_Search = document.getElementById("input2").value;
        var dataToSend = "?COMMAND=GETSEARCH&TAMCN=" + TAMCN_Search + "&NSN=" +NSN_Search + "&SERIAL_NUMBER="+ SERIAL_NUMBER_Search;
        req.open("GET", "http://172.20.151.182/examples/servlets/servlet/GetTheLocation" + dataToSend, true);
        req.onreadystatechange = handleServerResponse;
        req.send();
    }

    if (nsn_option ==1 && ser_option == 0){
        var dataToSend = "?COMMAND=GETSEARCH2&TAMCN=" + TAMCN_Search + "&NSN=" +NSN_Search ;
        req.open("GET", "http://172.20.151.182/examples/servlets/servlet/GetTheLocation" + dataToSend, true);
        req.onreadystatechange = handleServerResponse;
        req.send();
    }
}
```

Once this call reaches the server, it must then be parsed. In the case of a serial number search, the user's input (e.g., TAMCN, NSN, Serial Number) must be extracted after properly identifying the command being requested. Figure 11 shows some of the steps the server takes when communicating with the database for a serial number search. As the figure shows, a connection to the database is first made after identifying the command. Once this is complete, the server retrieves the user's input and places it into a

SQL statement. This query is then sent to the database and will return all information specific to that serial number. Because multiple tables in the database are utilized for the information returned to the user, it is critical that the consistency and integrity of the data is ensured. This is expanded upon later in the chapter, when discussing the Entity-Relationship diagram for the database schema.

Figure 11. Java Serial Number Search Command

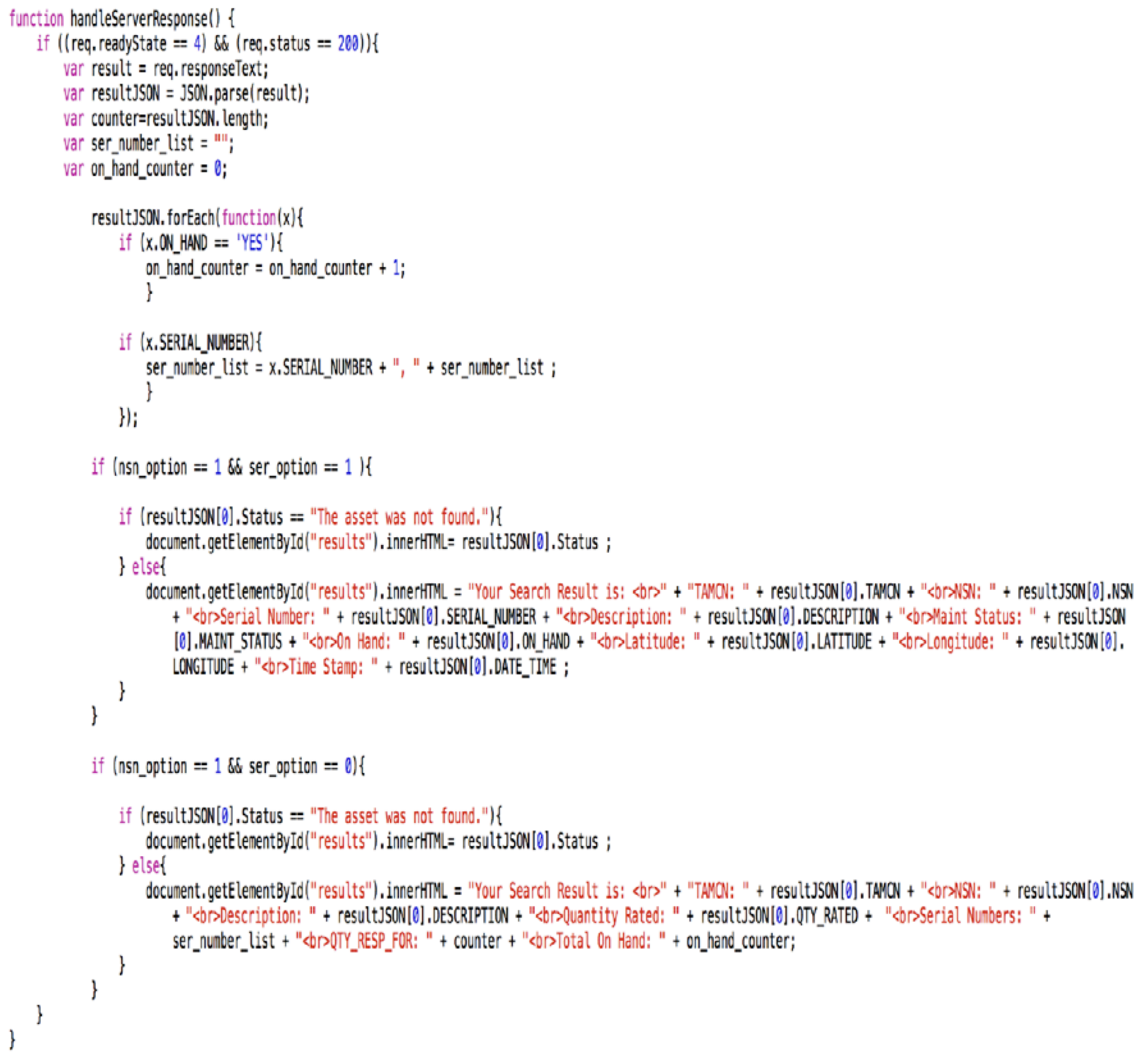


```
//------------------------------------ Search the Database (WITH S/N) ----------------------------------------//

    } else if (Command.equals("GETSEARCH")){

          try {
        Class.forName("oracle.jdbc.OracleDriver");
        System.out.println("Driver loaded");

        String url="jdbc:oracle:thin:@localhost:1521:xe";
        String user = "GCSS";
        String pwd = "password";

        Connection DB_mobile_conn = DriverManager.getConnection(url,user,pwd);
        System.out.println("Database Connect ok");
        String TAMCN= request.getParameter("TAMCN");
        String NSN= request.getParameter("NSN");
        String SERIAL_NUMBER= request.getParameter("SERIAL_NUMBER");

        String result = "TAMCN: " + TAMCN +" NSN: "+ NSN +" Serial Number: "+ SERIAL_NUMBER;
        System.out.println("result string : "+result);

        // Searches the database for the asset
        String searchQuery = "select t.*, a.DESCRIPTION from TRACKER_TABLE t, ASSET_TABLE a WHERE t.TAMCN='"+ TAMCN + "' AND t.NSN='" +
            NSN +"' AND t.SERIAL_NUMBER='" + SERIAL_NUMBER + "' AND t.TAMCN=a.TAMCN AND t.NSN=a.NSN";

        // Submits query and gets database
        Statement query_stmt = DB_mobile_conn.createStatement();
        ResultSet query_rs = query_stmt.executeQuery(searchQuery);

        ResultSetMetaData rsmd = query_rs.getMetaData();
        int queryColCount = rsmd.getColumnCount();
```

The response from the server back to the mobile device must be parsed from a JavaScript Object Notation (JSON) format in order to display the information on the mobile device's screen. Figure 12 details how the GCSS-MC mobile application handles the server response for each search reply. For a serial number search, the various elements in the JSON object simply need to be outputted onto the screen. With the more general NSN search, another step must be completed on the mobile device's part before the output can be displayed.

Two parts of the NSN search require a small computation and loop on the mobile device's side. These two outputs are the "Quantity Responsible For" and "Quantity On Hand" outcomes, and they require a counter in order for their results to be displayed as shown in Figure 12. In order to count the number of serial numbers of a NSN, simply using a length attribute of the JSON will suffice. However, determining the number of assets that are on hand requires a loop and counter. While these two computations are small, ideally this would be a server-based computation. In the interest of power-aware computing or being in a low bandwidth environment however, it is important to attempt to offload as much computation to the server.

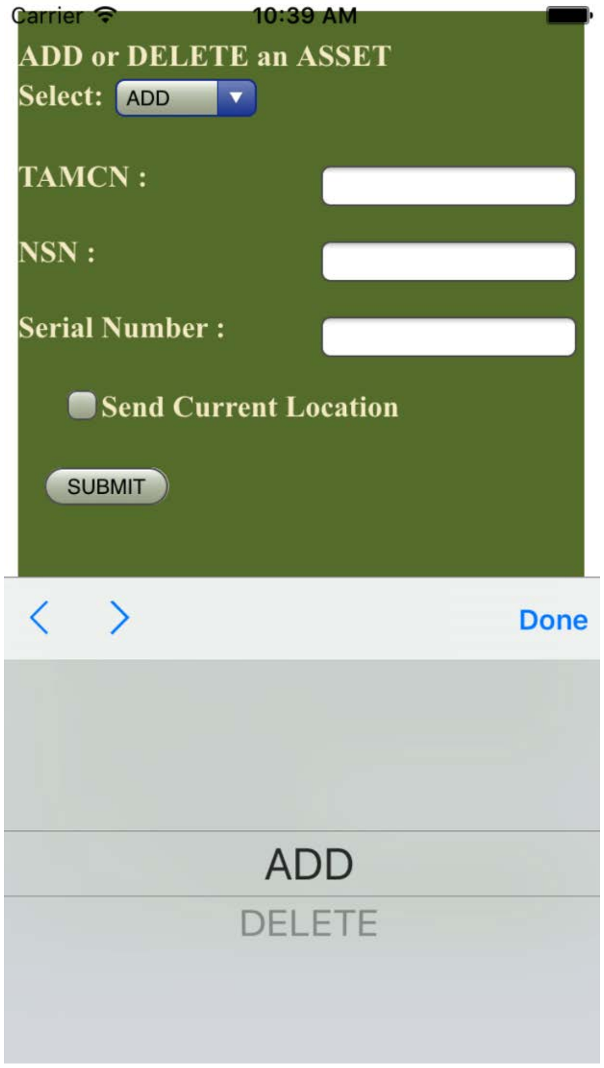
Figure 12. Displaying Search Results



```
function handleServerResponse() {
    if ((req.readyState == 4) && (req.status == 200)){
        var result = req.responseText;
        var resultJSON = JSON.parse(result);
        var counter=resultJSON.length;
        var ser_number_list = "";
        var on_hand_counter = 0;

            resultJSON.forEach(function(x){
                if (x.ON_HAND == 'YES'){
                    on_hand_counter = on_hand_counter + 1;
                    }

                if (x.SERIAL_NUMBER){
                    ser_number_list = x.SERIAL_NUMBER + ", " + ser_number_list ;
                    }
                });

            if (nsn_option == 1 && ser_option == 1 ){

                if (resultJSON[0].Status == "The asset was not found."){
                    document.getElementById("results").innerHTML= resultJSON[0].Status ;
                } else{
                    document.getElementById("results").innerHTML = "Your Search Result is: <br>" + "TAMCN: " + resultJSON[0].TAMCN + "<br>NSN: " + resultJSON[0].NSN
                        + "<br>Serial Number: " + resultJSON[0].SERIAL_NUMBER + "<br>Description: " + resultJSON[0].DESCRIPTION + "<br>Maint Status: " + resultJSON
                        [0].MAINT_STATUS + "<br>On Hand: " + resultJSON[0].ON_HAND + "<br>Latitude: " + resultJSON[0].LATITUDE + "<br>Longitude: " + resultJSON[0].
                        LONGITUDE + "<br>Time Stamp: " + resultJSON[0].DATE_TIME ;
                }
            }

            if (nsn_option == 1 && ser_option == 0){

                if (resultJSON[0].Status == "The asset was not found."){
                    document.getElementById("results").innerHTML= resultJSON[0].Status ;
                } else{
                    document.getElementById("results").innerHTML = "Your Search Result is: <br>" + "TAMCN: " + resultJSON[0].TAMCN + "<br>NSN: " + resultJSON[0].NSN
                        + "<br>Description: " + resultJSON[0].DESCRIPTION + "<br>Quantity Rated: " + resultJSON[0].QTY_RATED +  "<br>Serial Numbers: " +
                        ser_number_list + "<br>QTY_RESP_FOR: " + counter + "<br>Total On Hand: " + on_hand_counter;
                }
            }
    }
}
```

All communication between the user and database throughout the Property Module follows a structure similar to the interaction just described in the search configuration. Below the search section is the option for the user to add or delete an asset into the database. Figure 13 shows what the user will view when deciding whether to add or delete an item.

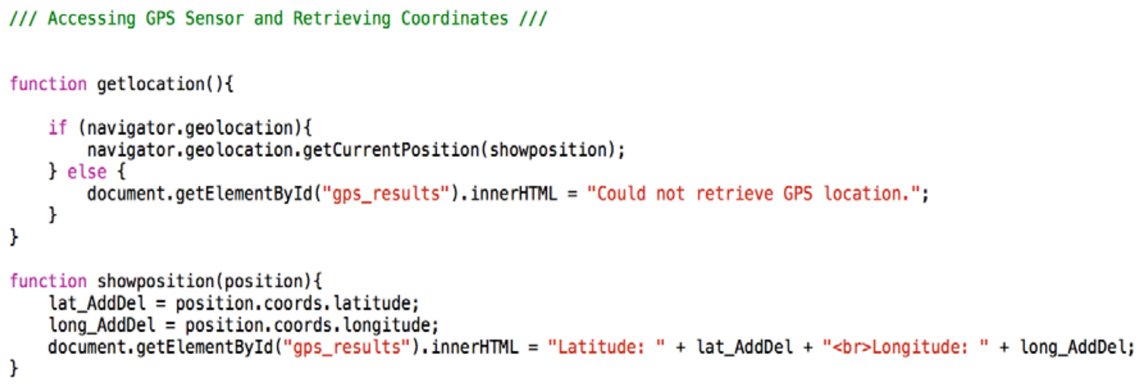
Figure 13. Add or Delete an Asset



Whether the user is attempting to add or delete an asset to or from the database, three requirements are needed: the TAMCN, NSN, and Serial Number. When adding an asset, however, the user is provided with an additional option to "Send Current Location." This grants the user the opportunity to send the GPS coordinates of the phone to the database. The most likely scenario for a user adding or deleting an asset from a mobile device will be when the user is located with the asset. For accountability purposes, associating the mobile device's current location to the asset would provide numerous benefits. Because this snapshot does not give a real time location however, a time stamp is also needed. At a minimum, this option would identify where that asset's exact location was at the time of the update. While the likely case is that the user is location with the asset at the time of update, this may not always be the case. Therefore, the user is given the flexibility to decide whether or not to send the current location to the database. An additional reason to grant the user this option is for security purposes. Some users may not be comfortable allowing the app to access the phone's GPS sensor or furthermore sending those coordinates to the database. Thus, a checkbox is provided to allow the user this option.

In order to access the phone's GPS and current coordinates, a simple function is created as shown in Figure 14. If the user grants permission for the sensor to be accessed, these functions in the figure will be called. The data they retrieve is sent to the database along with other needed information to properly add an asset to the GCSS-MC server.

Figure 14. GPS Functions

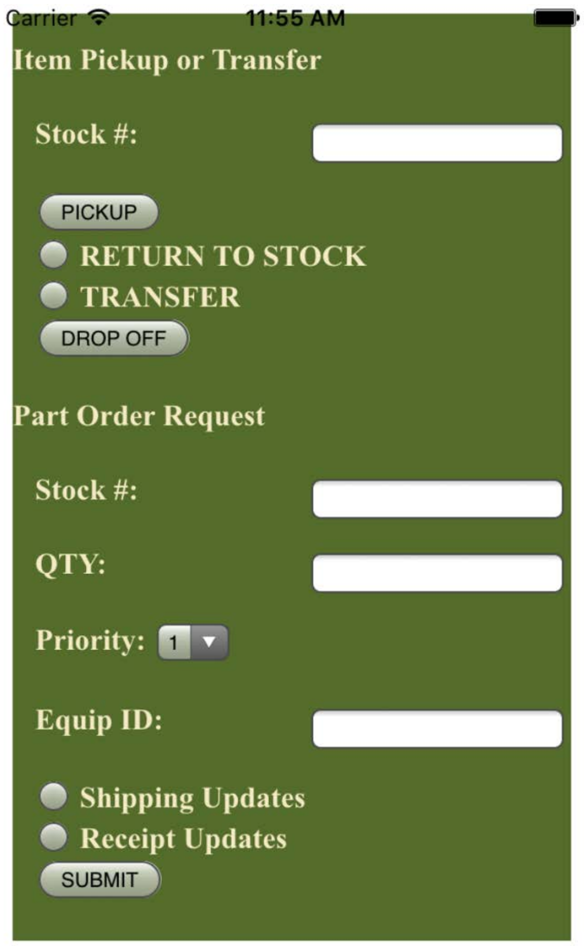


```
/// Accessing GPS Sensor and Retrieving Coordinates ///

function getlocation(){

    if (navigator.geolocation){
        navigator.geolocation.getCurrentPosition(showposition);
    } else {
        document.getElementById("gps_results").innerHTML = "Could not retrieve GPS location.";
    }
}

function showposition(position){
    lat_AddDel = position.coords.latitude;
    long_AddDel = position.coords.longitude;
    document.getElementById("gps_results").innerHTML = "Latitude: " + lat_AddDel + "<br>Longitude: " + long_AddDel;
}
```

The rest of the Property Module is not currently a working prototype. These functions fall outside the scope of the thesis and should be included in future research.

They do however provide a model of other capabilities that a typical GCSS-MC mobile user may want to utilize in the application. As viewed in Figure 15, an ability to administratively sign and annotate that item has been picked up or transferred would be beneficial. Changing the status of an item is a key part of ensuring supply functions operate efficiently. Returning items or transferring them to another section or unit occurs frequently. Having the ability to change the status of an asset by temporary loaning or returning the assets could improve current processes in the workplace.

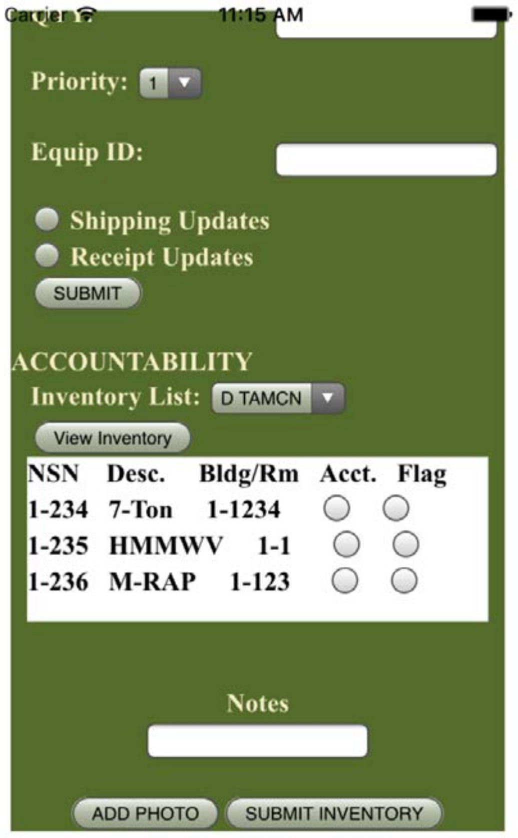
Figure 15. Item Pickup and Part Order Request



The second application found in Figure 15 is the ability to order parts. The defining feature from this section is the priority and update options. Being able to set priorities to parts requests allows the user to inform all stakeholders in the supply chain to know for instance if a submission is mission-critical. The update buttons will allow the user to receive various updates throughout the movement and who has physical and administrative control of an asset.

By scrolling to the very bottom of the Property Module as shown in Figure 16, the user will see the Accountability function of the application. The background of this thesis discusses the need for such a function and how it would improve the way Marines operate. There are numerous ways to present an inventory. In the format presented in Figure 16, the Inventory List of "D TAMCN"s is shown. "D TAMCN"s are a general term for motor transport vehicles in the Marine Corps. Once the "View Inventory" button is pressed, the D-TAMCNs under the responsibility of that CMR holder will be displayed.

Figure 16. Accountability Control Function



NSN	Desc.	Bldg/Rm	Acct.	Flag
1-234	7-Ton	1-1234	○	○
1-235	HMMWV	1-1	○	○
1-236	M-RAP	1-123	○	○

Once the user gets visual confirmation of a specific item, the user can then press the "Acct." radio button as seen on Figure 16 to annotate "eyes on" or "accounted for." Should there be any issue with this asset, the "Flag" radio button can also be pressed. This will identify to the user that there needs to be an adjustment of some sort on this particular item. It could mean that it is located in a different building as listed in the system or the NSN is inaccurate, for example. The notes feature, which is located just below the inventory, can be used to assist in documenting what specific changes need to be made.

The last feature in Figure 16 shows the "Add Photo" button. This will help the account holder provide proof in changes that may need to be made regarding the CMR, such as an incorrect serial number. A picture feature will also help users know exactly what that asset looks like, which is a tremendous benefit if employed effectively. These photos could be uploaded to a server that could be viewed at any given time.

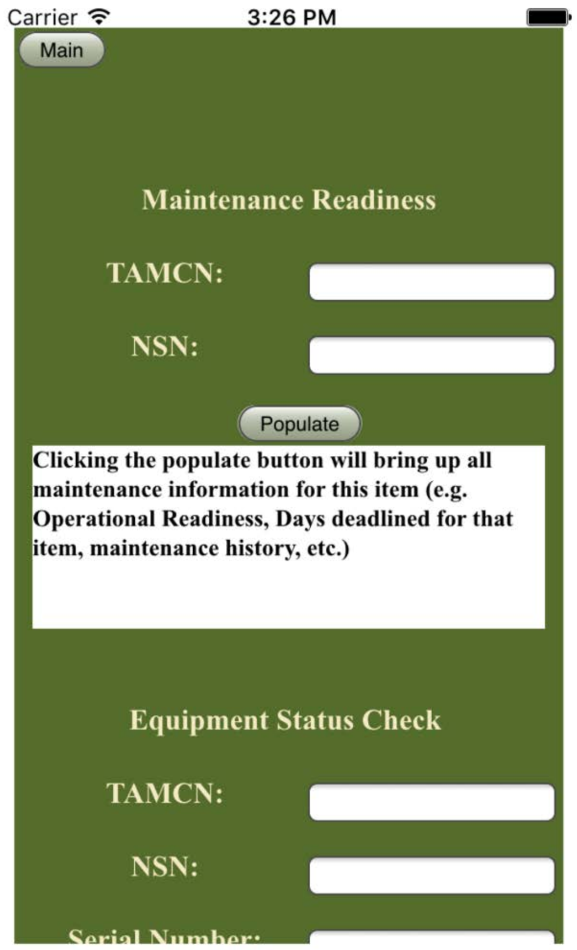
5. Maintenance Module

The last module of the GCSS-MC Mobile App is the Maintenance Module. The intent of this module is to provide all users who have a valid interest in the maintenance or operational readiness of the unit a means to perform administrative maintenance functions via a mobile device. Functions this module could accomplish range from keeping the user cognizant of the overall maintenance readiness of a unit to having the ability to order repair parts, and ability to update the maintenance status of an asset. The backend database connection and Java coding for this module falls outside the scope of this thesis and is recommended for future research.

The first function that the user views when opening the Maintenance Module is the Maintenance Readiness feature. The purpose of this section is to allow the user to enter information for a specific asset or general class of assets, such as all motor vehicles, and check their maintenance status, history, overall readiness, etc. As viewed in Figure 17, the first requirement for the user is to enter the data necessary for the servers to answer the requested query. This demands that the user know information, for instance the TAMCN and NSN, depending on the readiness specificity being requested. The

"Populate" button will then provide all maintenance information concerning that input. A wide array of relevant data regarding maintenance readiness can be shown in the output. For example, the operational readiness percentage of all motor vehicles under the user's responsibility could be shown as well as a listing of all items sorted by how long they have remained "deadlined" (not operational).

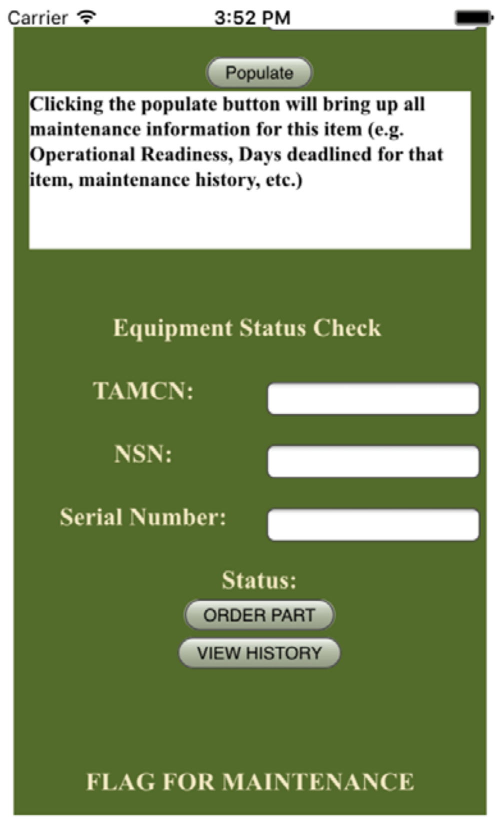
Figure 17. Maintenance Readiness Feature



Clicking the populate button will bring up all maintenance information for this item (e.g. Operational Readiness, Days deadlined for that item, maintenance history, etc.)

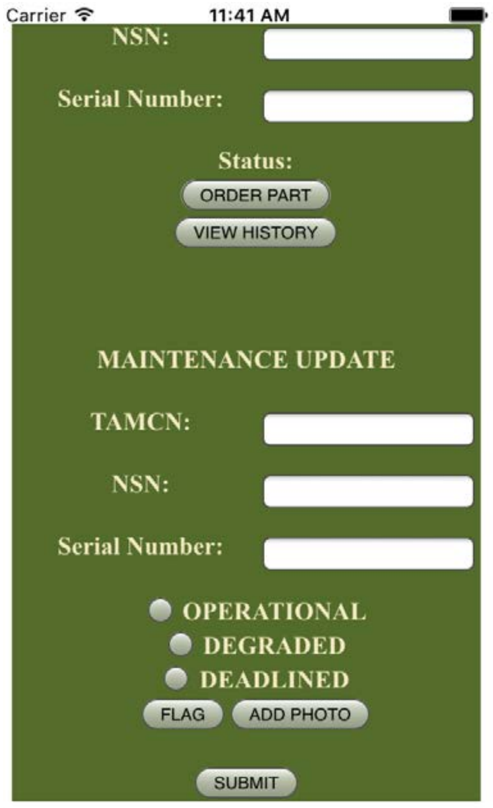
The second portion of the Maintenance Module is the Equipment Status Check feature. See Figure 18 for a screenshot for this section. Like the Maintenance Readiness segment the user will enter all pertinent information detailing that specific asset. Once complete, the user will be able to select two options. The first is a button that will provide a quick access ability to order repair parts for the piece of equipment needing maintenance. The “View History” button will grant the user the ability to check that specific assets current operational readiness and its history. For example, if a user would like to check how many times this individual asset has be in repair in the last year, the “View History” would not only provide this data, but state further details such as when, how long it was in repair, what the current or previous deficiencies are, and so on.

Figure 18. Equipment Status Check


Clicking the populate button will bring up all maintenance information for this item (e.g. Operational Readiness, Days deadlined for that item, maintenance history, etc.)

The last section in the Maintenance Module to be described is the Maintenance Update function, as shown in Figure 19. The user can reach this location by scrolling to the bottom of the module. This segment performs multiple functions to include allowing an asset to have it’s maintenance status updated, flagging an item for other pertinent remarks, and a photo feature for better describing a maintenance concern. To use these functions, all fields necessary to distinctly identify an item are required similar to the other sections in this module. After all required fields are properly entered into the prompt, the user can quickly update change the current maintenance status of an item by first selecting one of three options: Operational, Degraded, and Deadlined. After this is complete, the user may select the “Flag” button or “Add Photo” options. This is intended to provide further information regarding the maintenance status. The “Flag” button will allow the user to write a narrative, description, or provide other relevant material detailing the exact update being submitted. To further expand on this function the “Add Photo” button is available. This could be used to take a picture of the defective asset or specific item needing to be repaired or replaced. Employing this capability can be an invaluable practice for the Marine Corps if properly used. The photo feature would provide an innate quality control utility, as all stakeholders in the maintenance repair supply chain would be able to view the exact problem or update the GCSS-MC Mobile App user is submitting.

Figure 19. Maintenance Update Feature



Once all information is entered, the user can press the submit button in order to update the GCSS servers. Truly making a change on the servers would require that particular user has the required permissions to do so. If they do not allow him or her to make changes to the server, requests will be sent through the proper chain of command that would be able to.

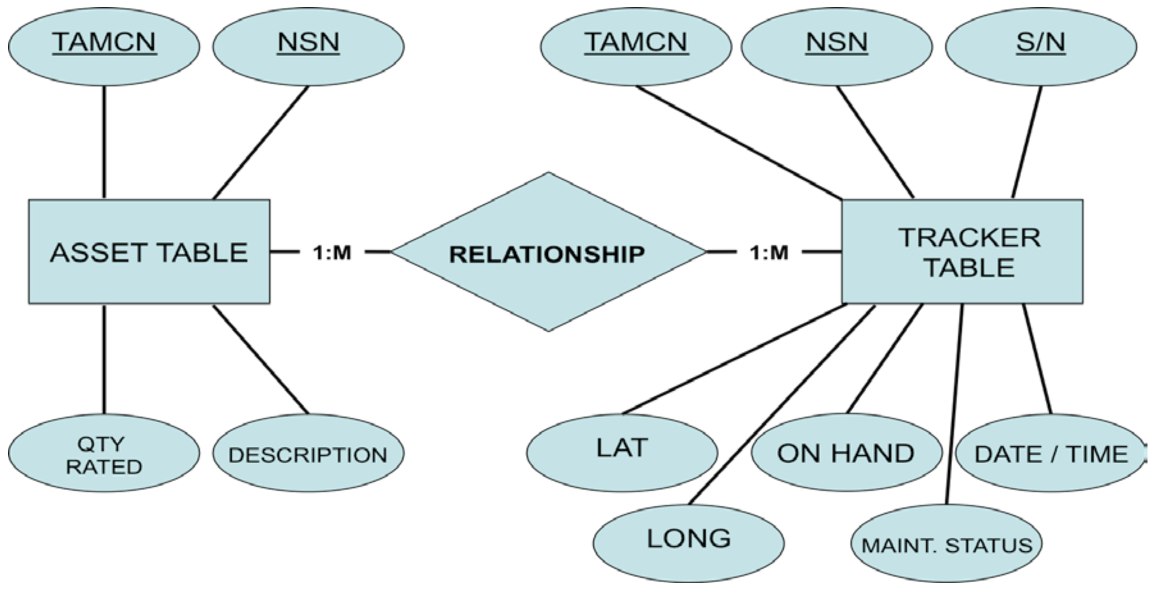
D. ENTITY RELATIONSHIP DIAGRAM

In order for any database to protect the consistency and integrity of its data, proper Entity-Relationship (E/R) diagrams can be used to help assure this occurs along with creating the conceptual modeling for the database. It is critical that the database administrator and program developer understand how the tables interrelate and how creating an E/R diagram assists in doing this. In order to translate an E/R diagram to the relational DBMS, there must also be a mapping between the two.

Figure 20 describes the relationship between two tables in the GCSS-MC mobile application. In this figure, the rectangles represent a relation or table in the database schema. The corresponding ovals will connect to form an entity. Each entity could also be described as a record in a database, while each oval represents a specific attribute of that relation. The underlined attributes formulate the key to that table. This key must also be unique in order to protect the integrity of the data.

Again by studying Figure 20, we consider the following examples. If a user simply requested the QTY RATED and DESCRIPTION of an asset, the database need only search in the ASSET TABLE after it has been providing the proper corresponding TAMCN and NSN. However, if a user had a more complex request, it may require the database to return data by referencing attributes in multiple tables. Had the user requested the description and location of a specific asset, two tables would need to be referenced. After the user provides the needed TAMCN, NSN, and S/N, the ASSET TABLE will return the DESCRIPTION, and the LAT and LONG will be pulled from the TRACKER TABLE to return the location.

Figure 20. Asset and Tracker Table E/R Diagram



While these are trivial queries compared to some of the more complex requests a user may have for GCSS-MC, they illustrate the value of data integrity and dependence in a well functioning database. Because the mobile application will be a subset or suite of the actual GCSS-MC program, the complexity of queries will easily be restrained at the developer's discretion. There are other considerations when assuring the data integrity of an enterprise-level database, but they fall out of the scope of this thesis.

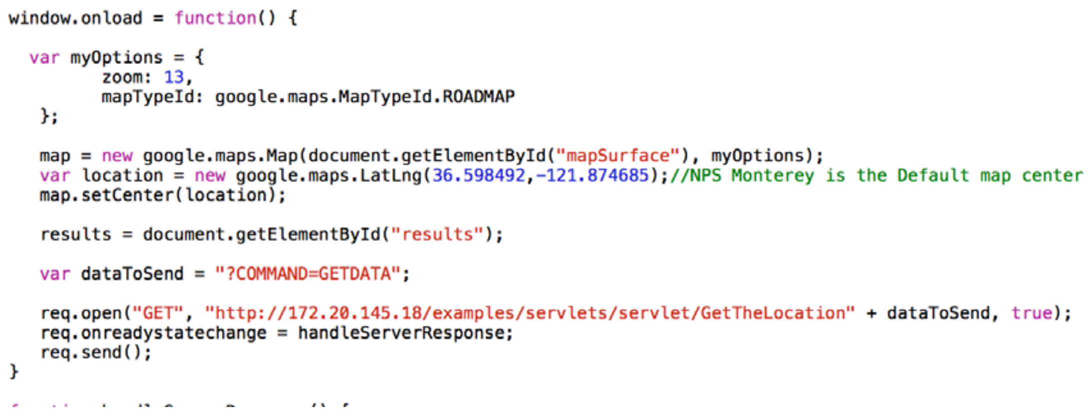
E. LOCATION SNAPSHOT

One of the most important features of the GCSS-MC mobile application is its ability to utilize the sensors on a mobile device. Many of which a laptop or desktop will not typically have. As mentioned when describing the Property Module of the application, the GCSS-MC mobile app can send a mobile device's coordinates to the database. To further extend this benefit, a webpage could be created to pinpoint these coordinates on a map and associate the asset to each point.

The first step in creating this feature is hosting a HTML webpage on a server that will be to present a map. The way it will be completed in this application is to first utilize

Google Maps' interactive maps. Because the testing will be completed at the Naval Postgraduate School in Monterey, CA, the map will be zoomed to this location to better visualize the pinpoints. See Figure 21 for a clear understanding of how this was accomplished.

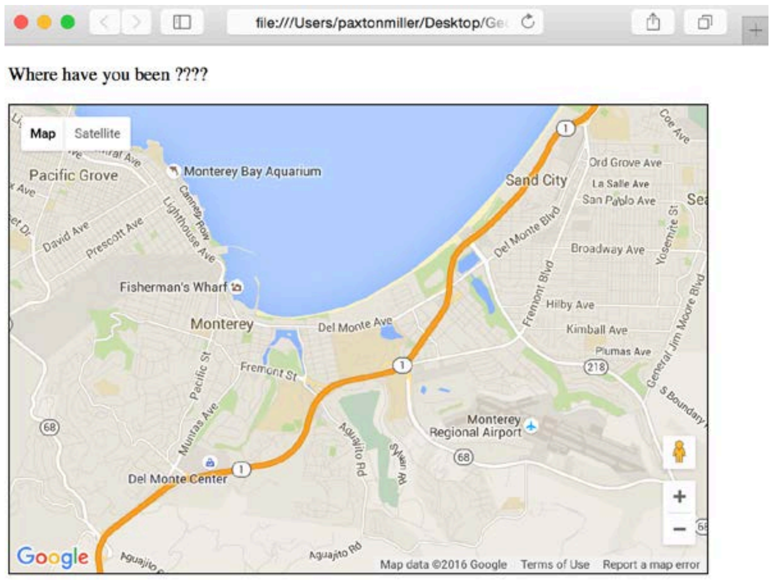
Figure 21. Map Presentation, Part 1



```
window.onload = function() {

  var myOptions = {
        zoom: 13,
        mapTypeId: google.maps.MapTypeId.ROADMAP
   };

   map = new google.maps.Map(document.getElementById("mapSurface"), myOptions);
   var location = new google.maps.LatLng(36.598492,-121.874685);//NPS Monterey is the Default map center
   map.setCenter(location);

   results = document.getElementById("results");

   var dataToSend = "?COMMAND=GETDATA";

   req.open("GET", "http://172.20.145.18/examples/servlets/servlet/GetTheLocation" + dataToSend, true);
   req.onreadystatechange = handleServerResponse;
   req.send();
}
```

Once the map is presented, the server requests the database server to provide the coordinates and other relevant data in order to pinpoint each asset's location. Prior to any assets being presented however, the webpage will appear as exhibited in Figure 22.

Figure 22. Map Presentation, Part 2

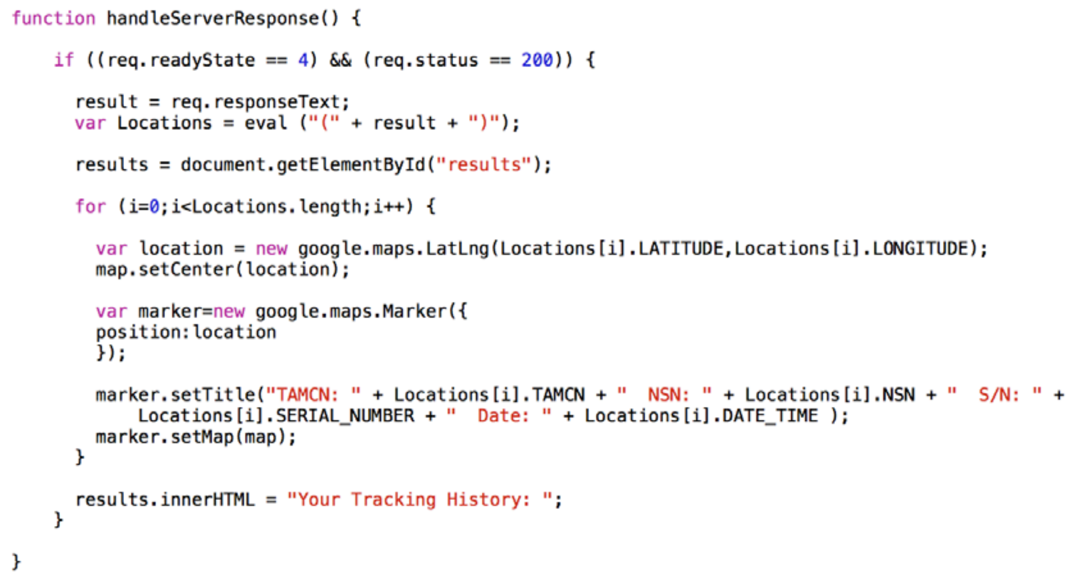


file:///Users/paxtonmiller/Desktop/Ge

Where have you been ????

Once assets have been correctly added, pinpoints that show the item’s serial number will be displayed. The way this is accomplished is by parsing through the data returned from the database server and presenting each asset’s information on the map. Figure 23 shows how the web server accomplishes this. A loop first pinpoints the coordinates provided from that element. From there, a “setTitle” marker is created that defines what that pinpoint designates. For the purposes of this thesis, TAMCN, NSN, Serial Number, description, and the time stamp of when the location was set is presented.

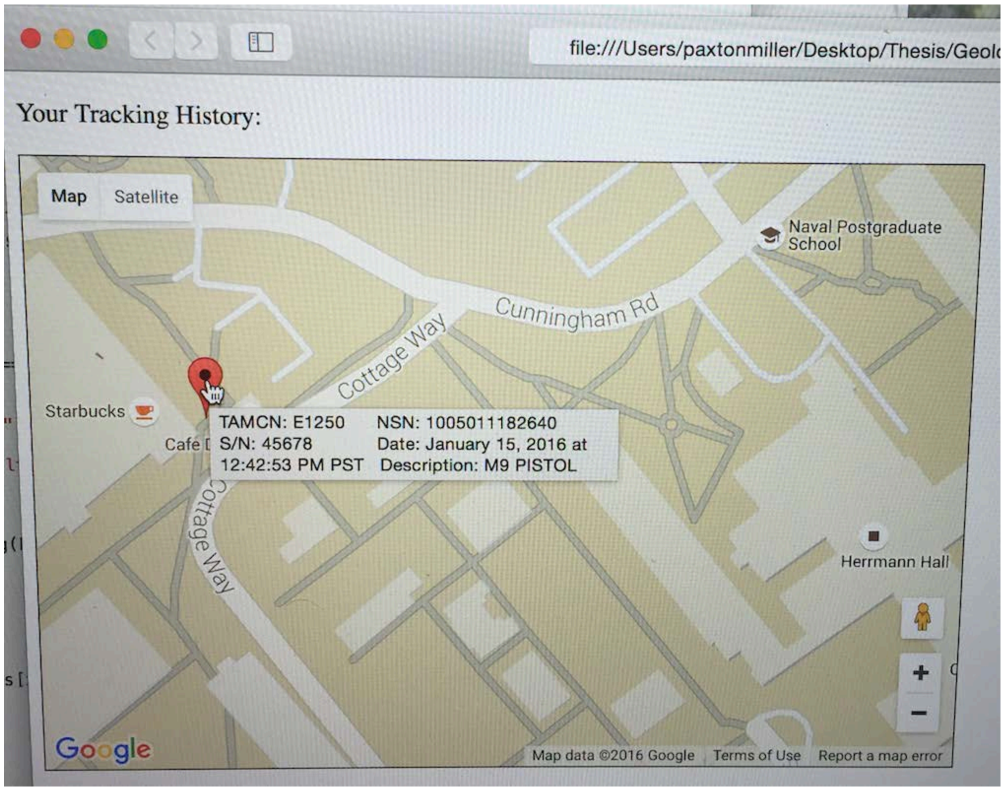
Figure 23. Map Presentation, Part 3



```
function handleServerResponse() {

    if ((req.readyState == 4) && (req.status == 200)) {

      result = req.responseText;
      var Locations = eval ("(" + result + ")");

      results = document.getElementById("results");

      for (i=0;i<Locations.length;i++) {

        var location = new google.maps.LatLng(Locations[i].LATITUDE,Locations[i].LONGITUDE);
        map.setCenter(location);

        var marker=new google.maps.Marker({
        position:location
        });

        marker.setTitle("TAMCN: " + Locations[i].TAMCN + "  NSN: " + Locations[i].NSN + "  S/N: " +
            Locations[i].SERIAL_NUMBER + "  Date: " + Locations[i].DATE_TIME );
        marker.setMap(map);
      }

      results.innerHTML = "Your Tracking History: ";
    }

}
```

In order to ensure that the database server and webpage were properly communicating, a record was hardcoded in the table with the location set to the Naval Postgraduate School. Once this was completed, the webpage was launched to assure that a pinpoint and a window describing it was being properly displayed. This is seen in Figure 24 after a M9 Pistol was manually entered into the database.

Figure 24. Map Presentation, Part 4



TAMCN: E1250 NSN: 1005011182640
S/N: 45678 Date: January 15, 2016 at
12:42:53 PM PST Description: M9 PISTOL

Chapter IV will describe the testing of the asset location snapshot feature, however the current webpage shows that interactive maps can easily be presented on webpage servers utilizing other existing capabilities, such as Google Maps. An obvious consideration for utilizing this feature is security and the authorization to access these technologies. The military cannot rely on Google Maps to present these points unless a contract allows it. Therefore, further research would be required to utilize bridging technologies to connect to existing tactical mapping software the military already has uploaded on DOD servers.

F. SUMMARY

Chapter III explored the GCSS-MC Mobile specifications, requirements, and architecture of what the application needs. Additionally, each module and how they were developed were also described. This includes the Profile Management, Property, and Maintenance modules. To better visualize this proof of concept application, screenshots of what these features could look like were also displayed throughout the chapter. After the application description was complete, the underlying database modeling it utilized was explained through the use of an Entity-Relationship Diagram. Finally, this chapter ends with a map feature that could further improve asset accountability across the Marine Corps through the benefit of the mobile app's GPS capabilities. Chapter IV addresses the testing of the mobile app and the map feature solutions.

IV. EVALUATION

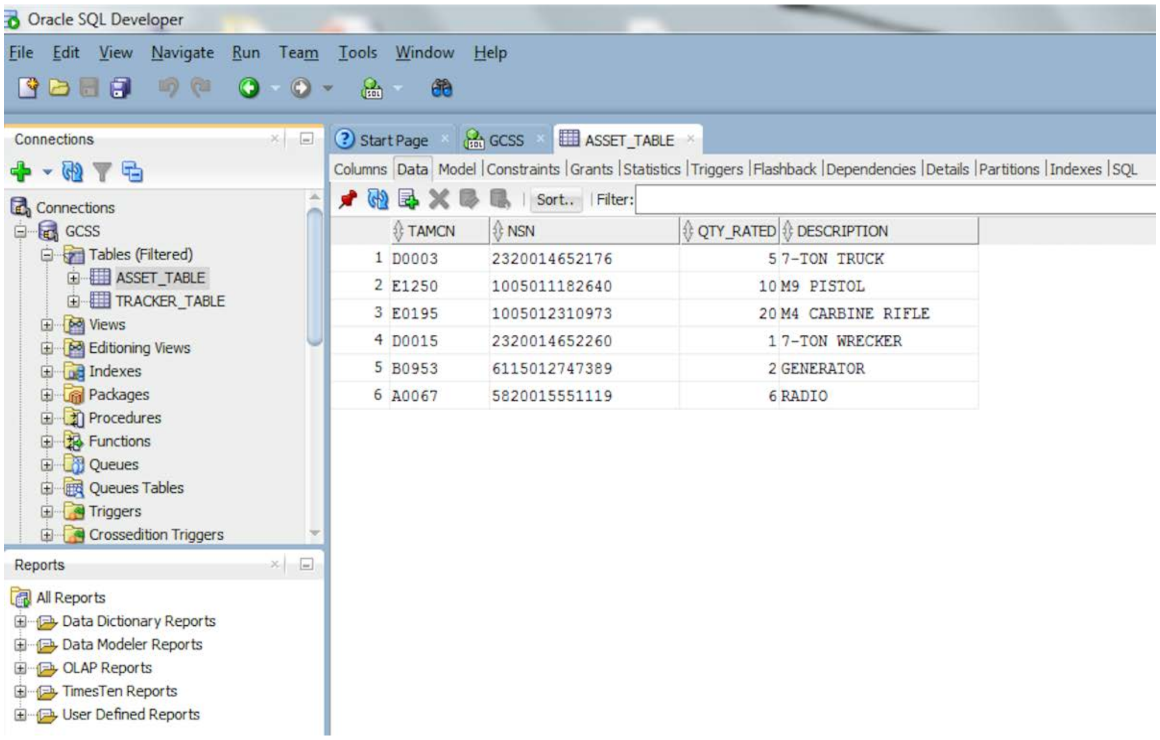
To properly test this application it was imperative to host it on an actual mobile device. While mobile device simulators, used from a laptop for instance, are beneficial, they do not provide the most suitable way to assess and evaluate the mobile application. Conducting the test in a real environment provides further credibility to the evaluation. This chapter evaluates various functions on the mobile application as well as the location snapshot webpage. The majority of the test was completed on an iPhone 6; however, an iPad Air was also employed to show that the application is usable and functional on a tablet as well.

A. APPLICATION TESTING

The application database utilizes two tables as described in ER Diagram portion of Chapter III. Those are the ASSET_TABLE and TRACKER_TABLE. Multiple tables are needed to normalize the database and organize it in a way that minimizes data redundancy. The ASSET_TABLE provides general information regarding military assets as shown in Figure 25. The attributes that make up this table are the TAMCN, NSN, the quantity of that asset rated by the user, and the nomenclature or description of the asset. While the ASSET_TABLE provides general information about assets, the TRACKER_TABLE is intended to provide data about a specific asset's serial number.

In order to establish the base information needed for the app to utilize for testing, data was entered into the ASSET_TABLE with the SQL Developer application. This was entered manually in the ASSET_TABLE of the app's database. While a user has the ability to make changes in the database, they are only authorized in the TRACKER_TABLE on an individual asset.

Figure 25. Asset Table

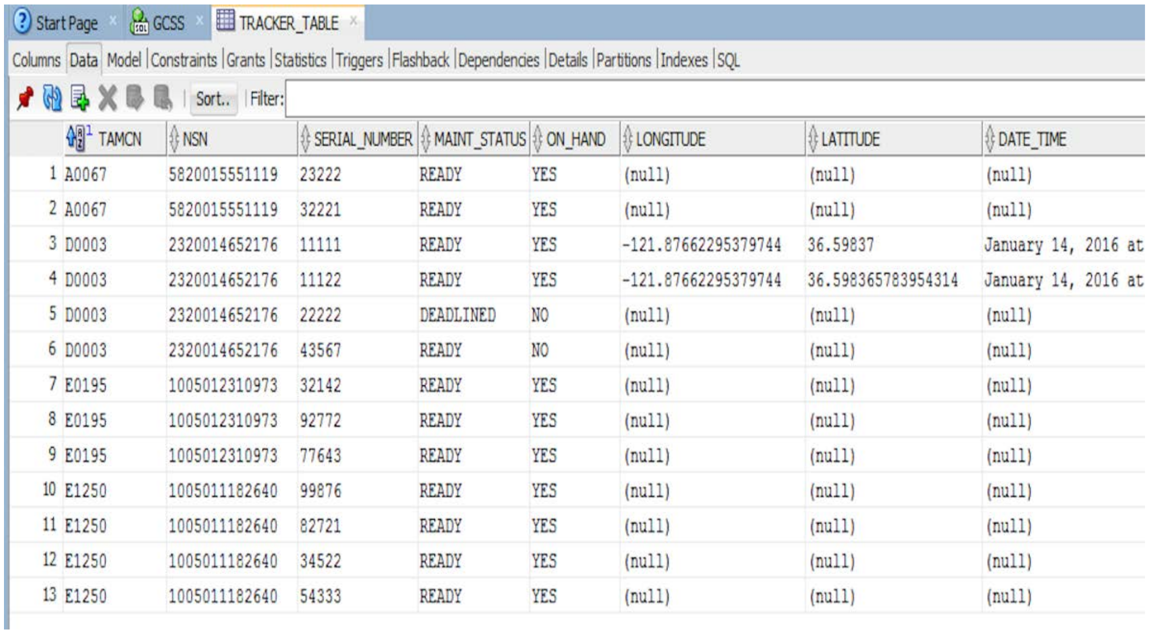


	TAMCN	NSN	QTY_RATED	DESCRIPTION
1	D0003	2320014652176	5	7-TON TRUCK
2	E1250	1005011182640	10	M9 PISTOL
3	E0195	1005012310973	20	M4 CARBINE RIFLE
4	D0015	2320014652260	1	7-TON WRECKER
5	B0953	6115012747389	2	GENERATOR
6	A0067	5820015551119	6	RADIO

After the base data was entered, the same process was completed on the Tracker Table, as shown in Figure 26. This was conducted for similar reasons that data was populated in the Asset Table. In order to manipulate data, such as completing an asset search or deleting an item, having base data for the phone will allow for a better application evaluation.

Not many of the assets hardcoded from the database server have the location or time stamp populated. This was intentional as a mobile device will be the primary tool for adding assets with this information during the evaluation. It should also be noted that because the TAMCN, NSN, and serial number form the primary key, the location and time stamp attributes are not needed for adding an asset. Once the tables have been created and the associated data populated, the application is ready to be launched.

Figure 26. Tracker Table


Start Page
GCSS
TRACKER_TABLE
Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL
Sort..
Filter:

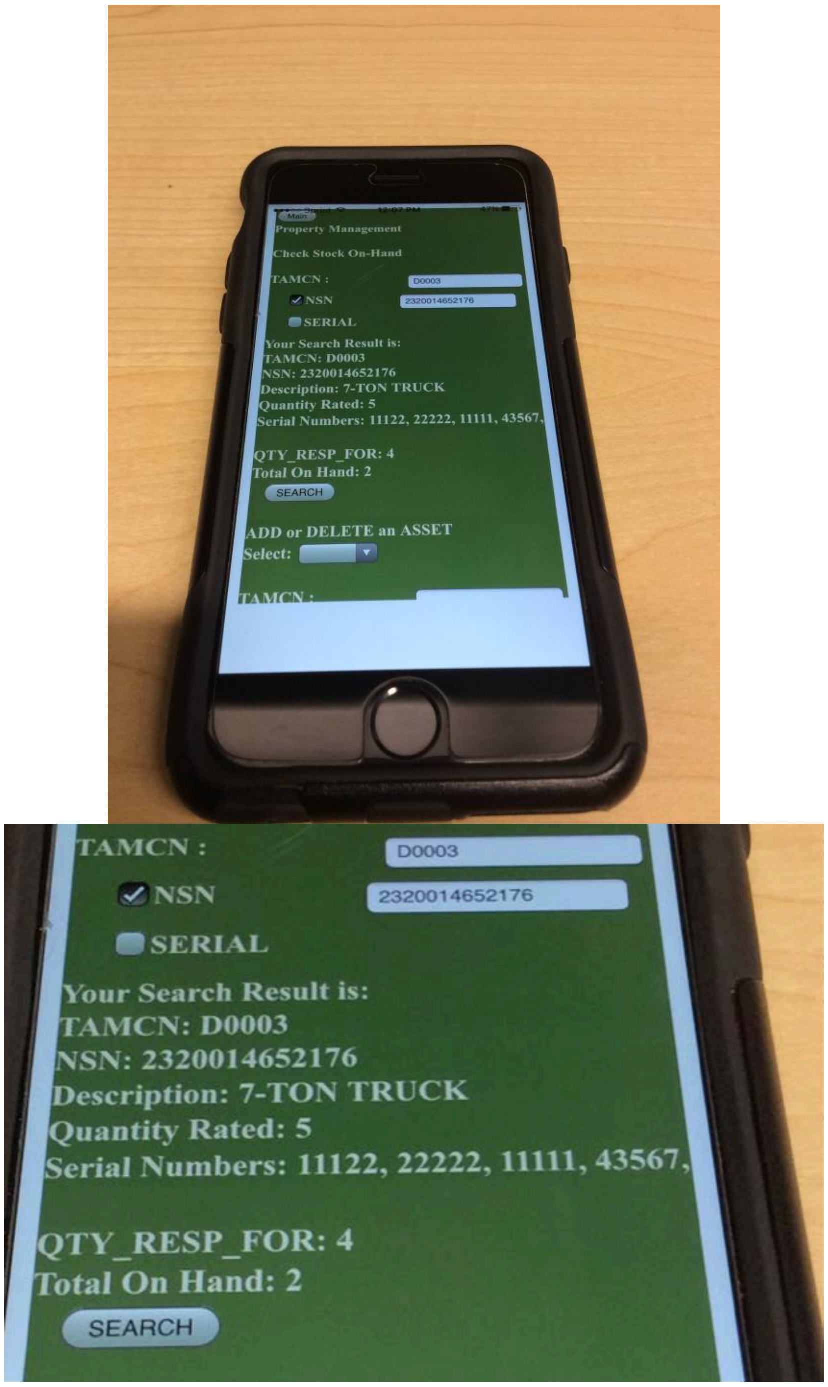
	TAMCN	NSN	SERIAL_NUMBER	MAINT_STATUS	ON_HAND	LONGITUDE	LATITUDE	DATE_TIME
1	A0067	5820015551119	23222	READY	YES	(null)	(null)	(null)
2	A0067	5820015551119	32221	READY	YES	(null)	(null)	(null)
3	D0003	2320014652176	11111	READY	YES	-121.87662295379744	36.59837	January 14, 2016 at
4	D0003	2320014652176	11122	READY	YES	-121.87662295379744	36.598365783954314	January 14, 2016 at
5	D0003	2320014652176	22222	DEADLINED	NO	(null)	(null)	(null)
6	D0003	2320014652176	43567	READY	NO	(null)	(null)	(null)
7	E0195	1005012310973	32142	READY	YES	(null)	(null)	(null)
8	E0195	1005012310973	92772	READY	YES	(null)	(null)	(null)
9	E0195	1005012310973	77643	READY	YES	(null)	(null)	(null)
10	E1250	1005011182640	99876	READY	YES	(null)	(null)	(null)
11	E1250	1005011182640	82721	READY	YES	(null)	(null)	(null)
12	E1250	1005011182640	34522	READY	YES	(null)	(null)	(null)
13	E1250	1005011182640	54333	READY	YES	(null)	(null)	(null)

1. Asset Search by NSN

All function tests on the app are completed in the Property Module. The first to be conducted is the NSN search. At the top of the module, the user enters the TAMCN into the prompt. Because TAMCNs are not unique and can have different NSNs, the user must also input the NSN in order for the database to know which asset is to be searched. Once this is complete and the search button is pressed, the search results returned are shown in Figure 27.

The results start by reiterating the input from the user, which are the TAMCN and NSN. Listed next is the description and quantity rated of the NSN. This information is returned from the ASSET_TABLE in the database. The next item in the search result is the listing of all serial numbers with that NSN under the user's responsibility. This data is pulled from the TRACKER_TABLE, where each serial number with the matching TAMCN and NSN is returned from the database to the mobile device. Following these returns, the user sees the "QTY_RESP_FOR" result. The number associated with this result is calculated by accumulating the number of serial numbers found in the search. In the example shown in Figure 27, the results inform the user that he or she is responsible for four 7-TON TRUCKS. The last element in the search result is the number of this NSN that is on hand or "Total On Hand." This output is calculated by collecting the total number of "YES"s in the ON_HAND column of the TRACKER_TABLE.

Figure 27. NSN Search Results Example



Property Management

Check Stock On-Hand

TAMCN : D0003

✓ NSN 2320014652176

SERIAL

Your Search Result is:
TAMCN: D0003
NSN: 2320014652176
Description: 7-TON TRUCK
Quantity Rated: 5
Serial Numbers: 11122, 22222, 11111, 43567,

QTY_RESP_FOR: 4
Total On Hand: 2

ADD or DELETE an ASSET
Select:

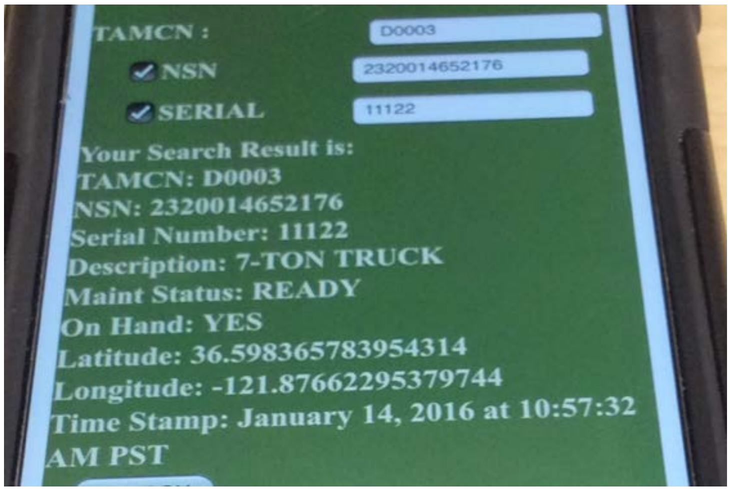
TAMCN :

By matching the information returned in Figure 27 to the data stored in the database, it is clear that the mobile device is receiving real-time and accurate results from the server for the NSN Search function.

2. Asset Search by Serial Number

The next function to be tested is the asset search by serial number. After the user selects the NSN and SERIAL checkboxes and fills in the prompts, all needed information to complete the search is complete. Once the search button is pressed, results similar to those found in Figure 28 are presented.

Figure 28. Serial Number Search Results Example

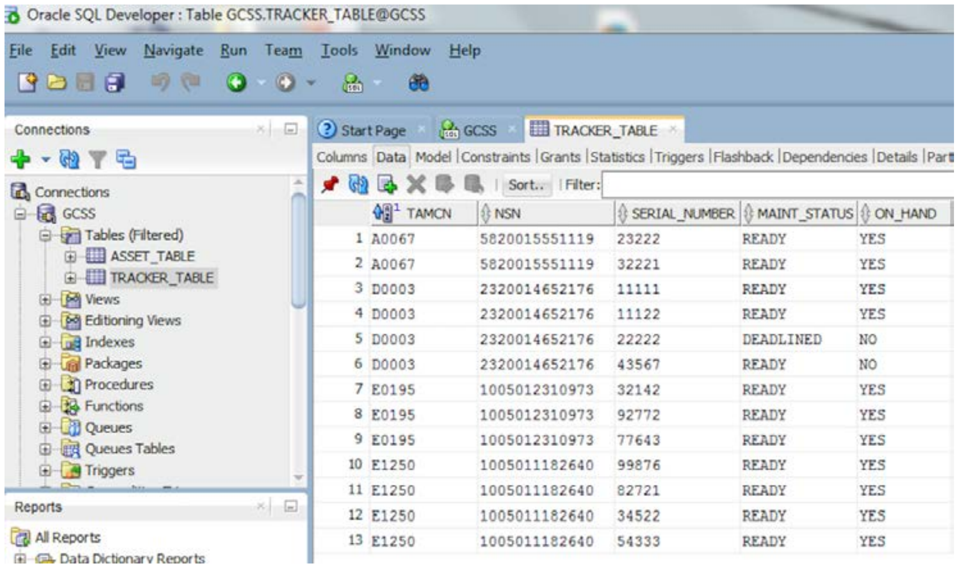

TAMCN : D0003
✓ NSN 2320014652176
✓ SERIAL 11122
Your Search Result is:
TAMCN: D0003
NSN: 2320014652176
Serial Number: 11122
Description: 7-TON TRUCK
Maint Status: READY
On Hand: YES
Latitude: 36.598365783954314
Longitude: -121.87662295379744
Time Stamp: January 14, 2016 at 10:57:32 AM PST

In Figure 28, serial number 11122 of a 7-TON TRUCK was searched. The results begin by detailing the TAMCN, NSN, serial number, and description of the asset al.l of this information was drawn from the ASSET_TABLE, while the remaining information was pulled from the TRACKER_TABLE of the database. Following the description search result, detailed data pertaining to that specific serial number is shown. This includes the maintenance status, the on-hand status, the last inputted GPS coordinates for that item, and the time stamp associated to that GPS input. The search results in Figure 28 match the data found in the database showing the successful functionality of the app's serial number search function.

3. Adding an Asset

The next function to test is the property module's ability to add an asset to the database. For this test, a generator with a TAMCN "B0953" and serial number "99990" will be added. To validate that this item is not already in the database, the data table is verified as shown in Figure 29.

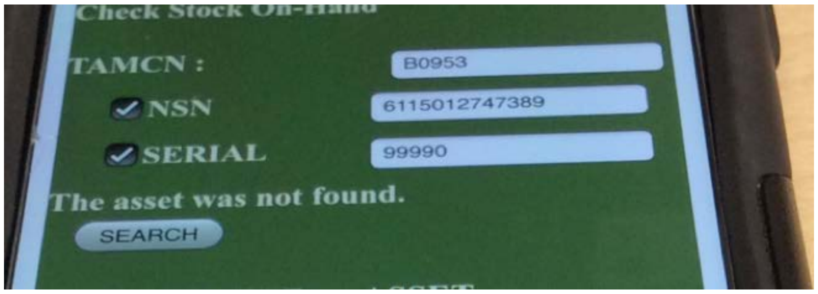
Figure 29. Adding an Asset, Part 1



	TAMCN	NSN	SERIAL_NUMBER	MAINT_STATUS	ON_HAND
1	A0067	5820015551119	23222	READY	YES
2	A0067	5820015551119	32221	READY	YES
3	D0003	2320014652176	11111	READY	YES
4	D0003	2320014652176	11122	READY	YES
5	D0003	2320014652176	22222	DEADLINED	NO
6	D0003	2320014652176	43567	READY	NO
7	E0195	1005012310973	32142	READY	YES
8	E0195	1005012310973	92772	READY	YES
9	E0195	1005012310973	77643	READY	YES
10	E1250	1005011182640	99876	READY	YES
11	E1250	1005011182640	82721	READY	YES
12	E1250	1005011182640	34522	READY	YES
13	E1250	1005011182640	54333	READY	YES

Next, in order to verify that the asset being adding is not already in the database from the user's perspective, the search function is used once again. After entering the needed information and the "Search" button is pressed, the query correctly returns with the statement "The asset was not found," as seen in Figure 30.

Figure 30. Adding an Asset, Part 2



TAMCN : B0953

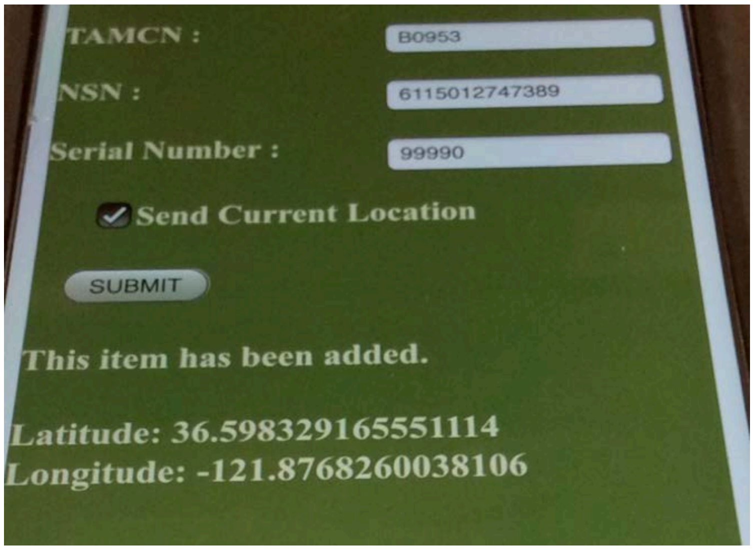
✓ NSN 6115012747389

✓ SERIAL 99990

The asset was not found.

After scrolling past the search function, the user may now enter the needed information for adding a generator. The user selects ADD from the “ADD or DELETE an ASSET” option, and the TAMCN, NSN, and serial number are entered into their respective prompts. Because the GPS feature also needs to be tested in this evaluation, the “Send Current Location” checkbox is also selected. Once this checkbox is populated, the user will be prompted by the phone to verify that it is acceptable for the application to capture the mobile device’s current location. Once this is allowed, the user may press the “Submit” button. The user will then receive a response similar to Figure 31, and a confirmation statement of “This item has been added.” will be displayed to the user.

Figure 31. Adding an Asset, Part 3



TAMCN : B0953

NSN : 6115012747389

Serial Number : 99990

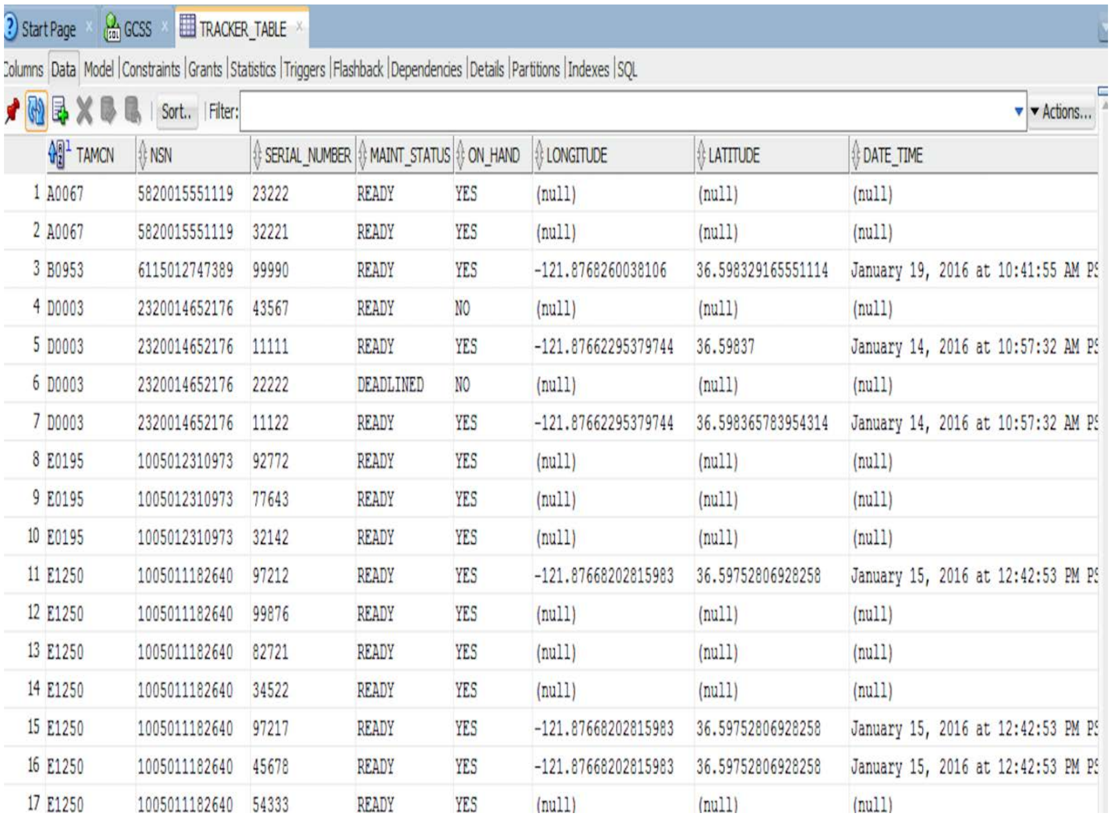
☑ Send Current Location

This item has been added.

Latitude: 36.598329165551114
Longitude: -121.8768260038106

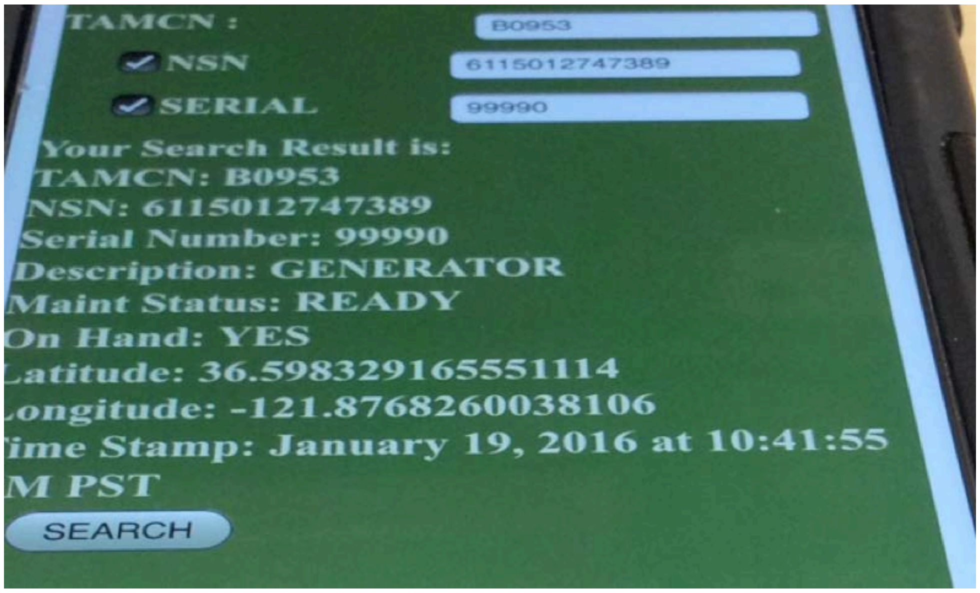
The last step in verifying that the generator has in fact been added is to return to the database server and the search function of the mobile device to locate the newly added asset. Record 3 of Figure 32 shows that not only has the item been added to the database, but that the GPS coordinates and time stamp was also captured. The search function on the mobile device was used once again with the generator’s asset information, and the results are shown in Figure 33. It shows that the ADD function was successful from the mobile device’s perspective as well.

Figure 32. Adding an Asset, Part 4



	TAMCN	NSN	SERIAL_NUMBER	MAINT_STATUS	ON_HAND	LONGITUDE	LATITUDE	DATE_TIME
1	A0067	5820015551119	23222	READY	YES	(null)	(null)	(null)
2	A0067	5820015551119	32221	READY	YES	(null)	(null)	(null)
3	B0953	6115012747389	99990	READY	YES	-121.8768260038106	36.598329165551114	January 19, 2016 at 10:41:55 AM PS
4	D0003	2320014652176	43567	READY	NO	(null)	(null)	(null)
5	D0003	2320014652176	11111	READY	YES	-121.87662295379744	36.59837	January 14, 2016 at 10:57:32 AM PS
6	D0003	2320014652176	22222	DEADLINED	NO	(null)	(null)	(null)
7	D0003	2320014652176	11122	READY	YES	-121.87662295379744	36.598365783954314	January 14, 2016 at 10:57:32 AM PS
8	E0195	1005012310973	92772	READY	YES	(null)	(null)	(null)
9	E0195	1005012310973	77643	READY	YES	(null)	(null)	(null)
10	E0195	1005012310973	32142	READY	YES	(null)	(null)	(null)
11	E1250	1005011182640	97212	READY	YES	-121.87668202815983	36.59752806928258	January 15, 2016 at 12:42:53 PM PS
12	E1250	1005011182640	99876	READY	YES	(null)	(null)	(null)
13	E1250	1005011182640	82721	READY	YES	(null)	(null)	(null)
14	E1250	1005011182640	34522	READY	YES	(null)	(null)	(null)
15	E1250	1005011182640	97217	READY	YES	-121.87668202815983	36.59752806928258	January 15, 2016 at 12:42:53 PM PS
16	E1250	1005011182640	45678	READY	YES	-121.87668202815983	36.59752806928258	January 15, 2016 at 12:42:53 PM PS
17	E1250	1005011182640	54333	READY	YES	(null)	(null)	(null)

Figure 33. Adding an Asset, Part 5



TAMCN : B0953

✓ NSN 6115012747389

✓ SERIAL 99990

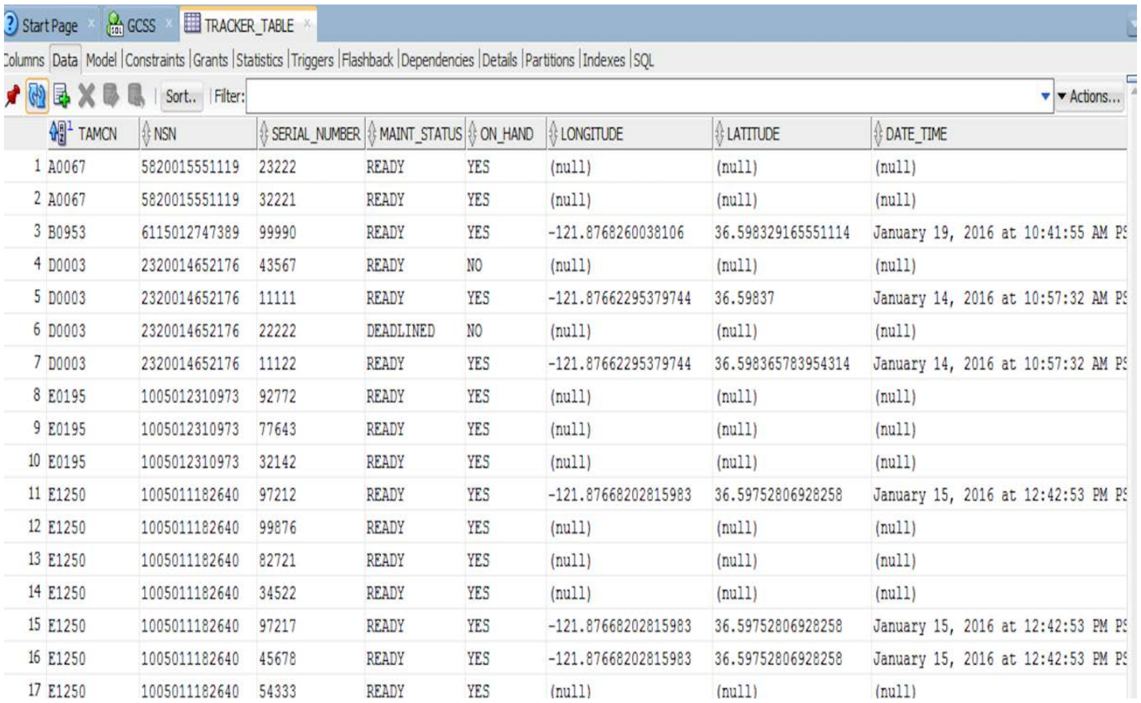
Your Search Result is:
TAMCN: B0953
NSN: 6115012747389
Serial Number: 99990
Description: GENERATOR
Maint Status: READY
On Hand: YES
Latitude: 36.598329165551114
Longitude: -121.8768260038106
Time Stamp: January 19, 2016 at 10:41:55
M PST

It was key for the verification from the phone’s perspective to be successful as well, because not only does it give the user confirmation that the change was made to the database, but also that any other user of GCSS-MC or its mobile app will see this update to the database in real time.

4. Deleting an Asset

The next function to test is deleting an asset from the database. The way this will be tested is by identifying an asset in the database along with its associated serial number, and removing that item through the mobile device. In this example, a radio with a TAMCN of A0067 and serial number of 23222 is identified as the asset to be removed. This is seen as the first record in Figure 34.

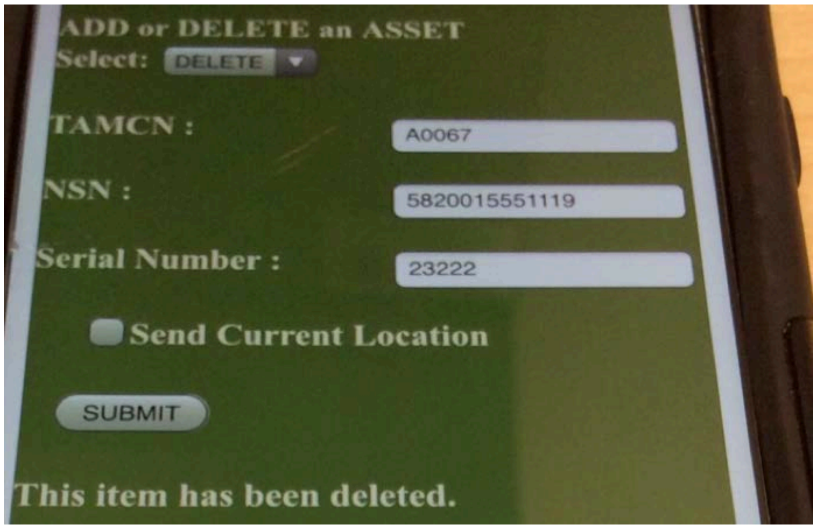
Figure 34. Deleting an Asset, Part 1



	TAMCN	NSN	SERIAL_NUMBER	MAINT_STATUS	ON_HAND	LONGITUDE	LATITUDE	DATE_TIME
1	A0067	5820015551119	23222	READY	YES	(null)	(null)	(null)
2	A0067	5820015551119	32221	READY	YES	(null)	(null)	(null)
3	B0953	6115012747389	99990	READY	YES	-121.8768260038106	36.598329165551114	January 19, 2016 at 10:41:55 AM PS
4	D0003	2320014652176	43567	READY	NO	(null)	(null)	(null)
5	D0003	2320014652176	11111	READY	YES	-121.87662295379744	36.59837	January 14, 2016 at 10:57:32 AM PS
6	D0003	2320014652176	22222	DEADLINED	NO	(null)	(null)	(null)
7	D0003	2320014652176	11122	READY	YES	-121.87662295379744	36.598365783954314	January 14, 2016 at 10:57:32 AM PS
8	E0195	1005012310973	92772	READY	YES	(null)	(null)	(null)
9	E0195	1005012310973	77643	READY	YES	(null)	(null)	(null)
10	E0195	1005012310973	32142	READY	YES	(null)	(null)	(null)
11	E1250	1005011182640	97212	READY	YES	-121.87668202815983	36.59752806928258	January 15, 2016 at 12:42:53 PM PS
12	E1250	1005011182640	99876	READY	YES	(null)	(null)	(null)
13	E1250	1005011182640	82721	READY	YES	(null)	(null)	(null)
14	E1250	1005011182640	34522	READY	YES	(null)	(null)	(null)
15	E1250	1005011182640	97217	READY	YES	-121.87668202815983	36.59752806928258	January 15, 2016 at 12:42:53 PM PS
16	E1250	1005011182640	45678	READY	YES	-121.87668202815983	36.59752806928258	January 15, 2016 at 12:42:53 PM PS
17	E1250	1005011182640	54333	READY	YES	(null)	(null)	(null)

Next, the user will navigate to the “ADD or DELETE an ASSET” scrollbar on the mobile device. Then, the user will select the DELETE option from this scrollbar. Like adding an asset, the user will subsequently input the needed information in order to make the requested database adjustment. After this is complete, the user may press the "Submit" button. Figure 35 shows the phones results after deleting the TAMCN A0067 with serial number 23222. Like adding an asset to the database, the user will receive a confirmation response with the statement "This item has been deleted."

Figure 35. Deleting an Asset, Part 2


ADD or DELETE an ASSET

Select: DELETE

TAMCN : A0067

NSN : 5820015551119

Serial Number : 23222

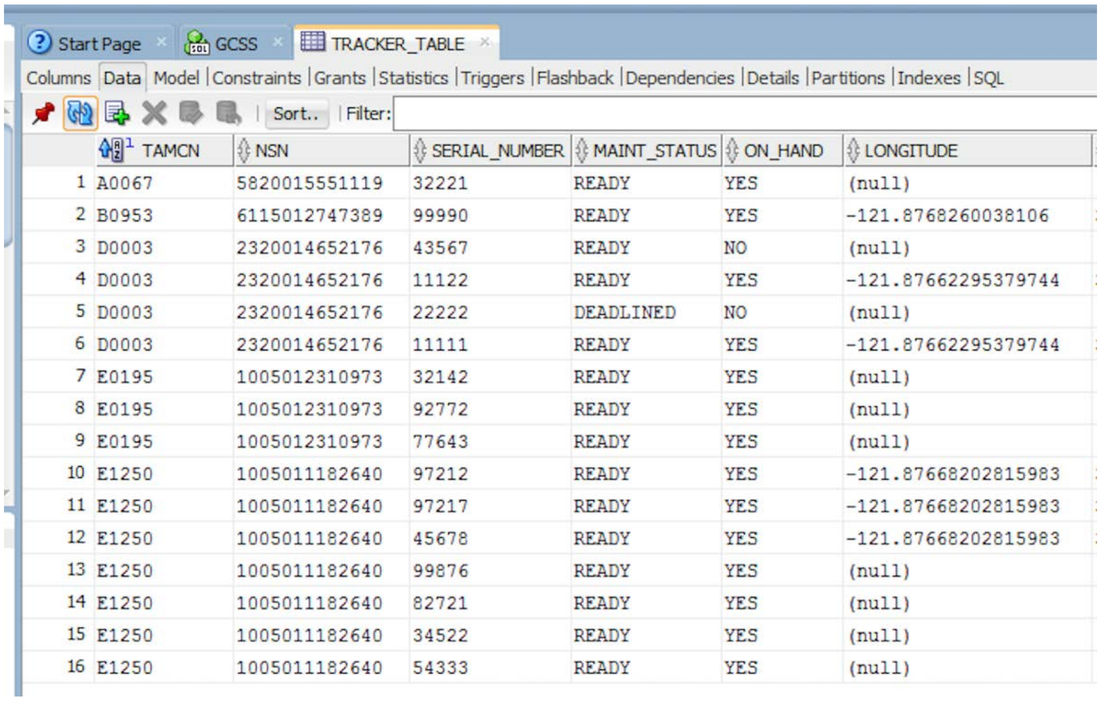
Send Current Location

SUBMIT

This item has been deleted.

Following the deletion of this asset, the database is refreshed to view the updated change. As shown in Figure 36, the record for serial number 23222 has been deleted from the database server and is no longer the first record in the TRACKER_TABLE. This change validates that the mobile device DELETE feature is properly functioning on the mobile device.

Figure 36. Deleting an Asset, Part 3



Start Page
GCSS
TRACKER_TABLE
Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL
Sort..
Filter:

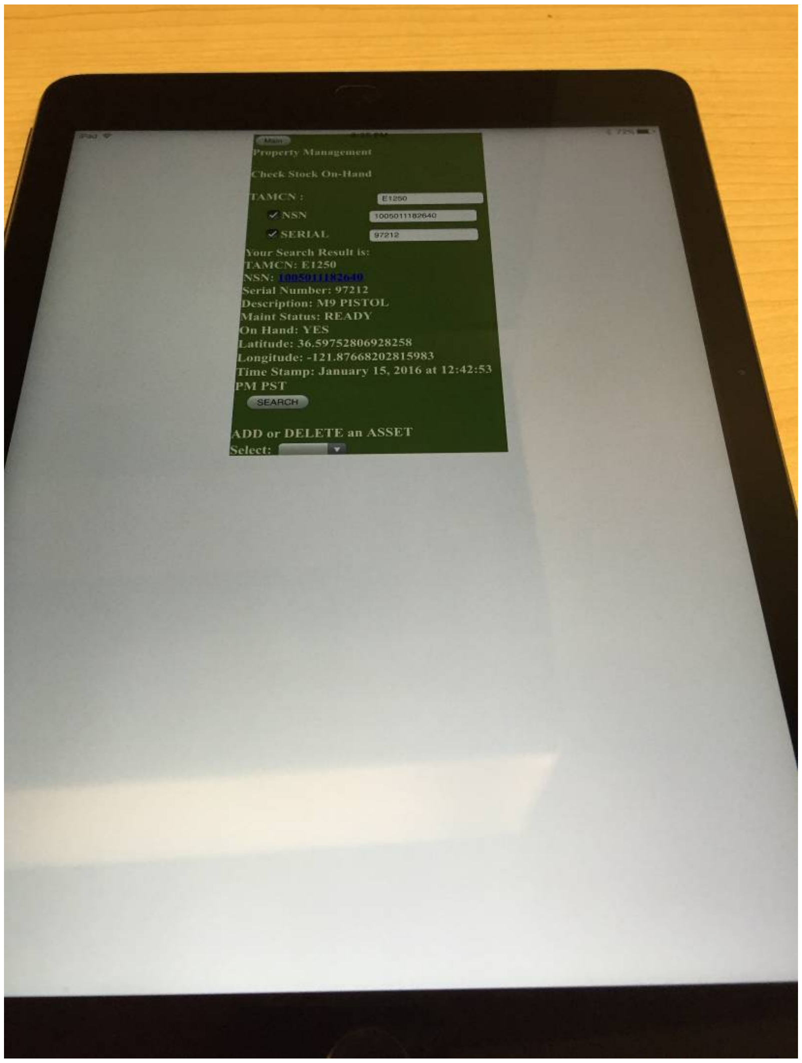
	TAMCN	NSN	SERIAL_NUMBER	MAINT_STATUS	ON_HAND	LONGITUDE
1	A0067	5820015551119	32221	READY	YES	(null)
2	B0953	6115012747389	99990	READY	YES	-121.8768260038106
3	D0003	2320014652176	43567	READY	NO	(null)
4	D0003	2320014652176	11122	READY	YES	-121.87662295379744
5	D0003	2320014652176	22222	DEADLINED	NO	(null)
6	D0003	2320014652176	11111	READY	YES	-121.87662295379744
7	E0195	1005012310973	32142	READY	YES	(null)
8	E0195	1005012310973	92772	READY	YES	(null)
9	E0195	1005012310973	77643	READY	YES	(null)
10	E1250	1005011182640	97212	READY	YES	-121.87668202815983
11	E1250	1005011182640	97217	READY	YES	-121.87668202815983
12	E1250	1005011182640	45678	READY	YES	-121.87668202815983
13	E1250	1005011182640	99876	READY	YES	(null)
14	E1250	1005011182640	82721	READY	YES	(null)
15	E1250	1005011182640	34522	READY	YES	(null)
16	E1250	1005011182640	54333	READY	YES	(null)

The remaining features discussed in Chapter III are not currently functional and are recommended for future research. These tasks that were evaluated, however, do prove the successful implementation of the functions that fall within the scope of this thesis.

B. TABLET TESTING

The intent of testing the application on a tablet is simply to show that the GCSS-MC mobile application is functioning properly on both types of mobile devices. The type of tablet used for this evaluation is an iPad Air. In order to simply show that the mobile application works on a tablet, a simple search on an asset in the database will be completed. After launching the app and performing the function as shown in Figure 37, it becomes clear that formatting needs adjustment to utilize the entire screen on a tablet. From a functionality point of view, the figure does however show that the functions do work. While amending the application to account for all types of mobile devices from a formatting standpoint would be beneficial, it does show that the application is runs correctly on tablets and meets the success criteria of the features that have been developed.

Figure 37. Tablet Testing Example



Property Management

Check Stock On-Hand

TAMCN : E1250

✓ NSN 1005011182640

✓ SERIAL 97212

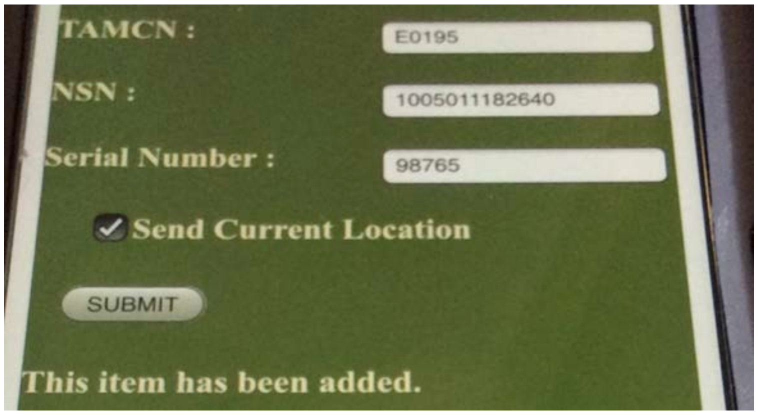
Your Search Result is:
TAMCN: E1250
NSN: 1005011182640
Serial Number: 97212
Description: M9 PISTOL
Maint Status: READY
On Hand: YES
Latitude: 36.59752806928258
Longitude: -121.87668202815983
Time Stamp: January 15, 2016 at 12:42:53 PM PST

ADD or DELETE an ASSET
Select:

C. LOCATION SNAPSHOT TESTING

In order to show that the location snapshot webpage is properly working, multiple assets around the Naval Postgraduate School campus were added to the database with the location checkbox turned on. For testing this feature, one asset specifically will be highlighted. A M4 Rifle with serial number 98765 will first be added to the database. Once the information has been properly submitted, the user is notified that the asset has been added, as shown in Figure 38.

Figure 38. Location Snapshot, Part 1



TAMCN : E0195

NSN : 1005011182640

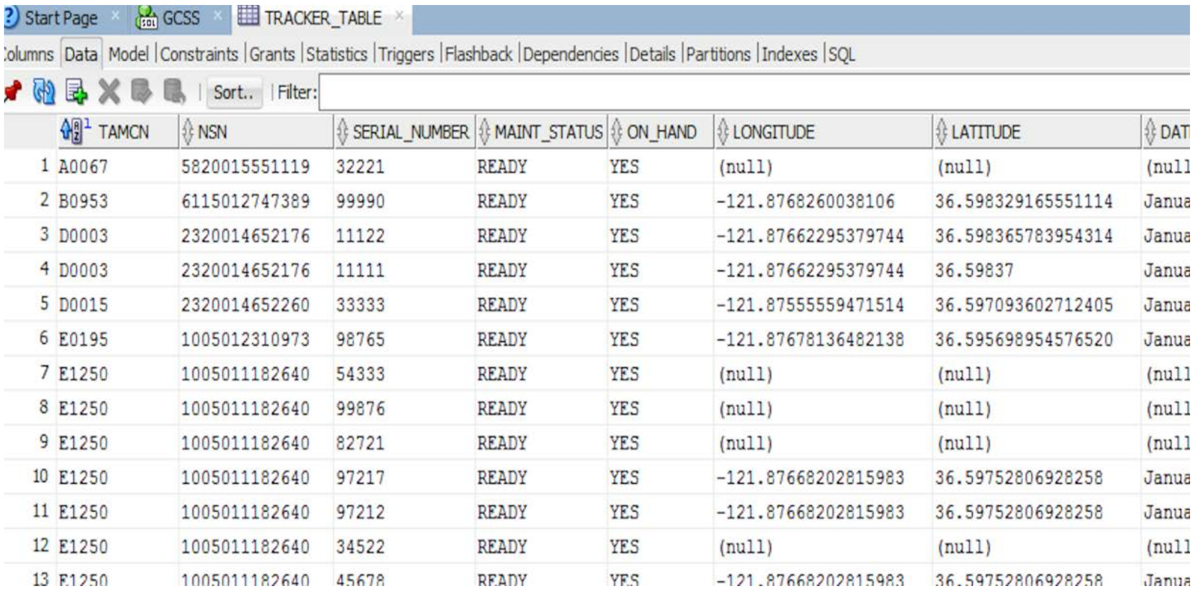
Serial Number : 98765

✓ Send Current Location

This item has been added.

After returning to the server and reviewing the assets in the database, the recently added M4 Rifle with serial number 98765 is found. See record 6 in Figure 39 as confirmation.

Figure 39. Location Snapshot, Part 2

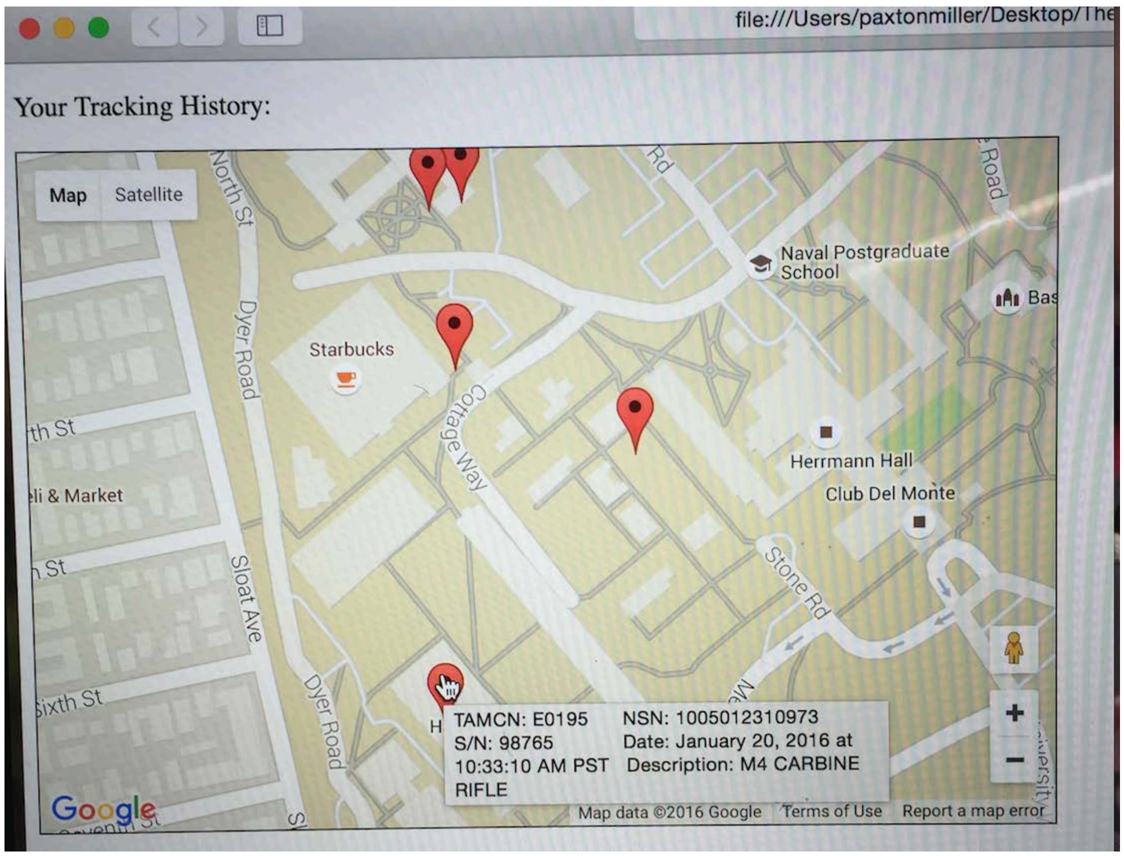


Start Page
GCSS
TRACKER_TABLE
olumns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL
Sort..
Filter:

	TAMCN	NSN	SERIAL_NUMBER	MAINT_STATUS	ON_HAND	LONGITUDE	LATITUDE	DAT
1	A0067	5820015551119	32221	READY	YES	(null)	(null)	(null
2	B0953	6115012747389	99990	READY	YES	-121.8768260038106	36.598329165551114	Janua
3	D0003	2320014652176	11122	READY	YES	-121.87662295379744	36.598365783954314	Janua
4	D0003	2320014652176	11111	READY	YES	-121.87662295379744	36.59837	Janua
5	D0015	2320014652260	33333	READY	YES	-121.87555559471514	36.597093602712405	Janua
6	E0195	1005012310973	98765	READY	YES	-121.87678136482138	36.595698954576520	Janua
7	E1250	1005011182640	54333	READY	YES	(null)	(null)	(null
8	E1250	1005011182640	99876	READY	YES	(null)	(null)	(null
9	E1250	1005011182640	82721	READY	YES	(null)	(null)	(null
10	E1250	1005011182640	97217	READY	YES	-121.87668202815983	36.59752806928258	Janua
11	E1250	1005011182640	97212	READY	YES	-121.87668202815983	36.59752806928258	Janua
12	E1250	1005011182640	34522	READY	YES	(null)	(null)	(null
13	E1250	1005011182640	45678	READY	YES	-121.87668202815983	36.59752806928258	Janua

Once this has been completed, the Location Snapshot webpage is opened to view the assets that have been added during the evaluation. After opening the page, pinpoints around the Naval Postgraduate School campus are shown. Once the user moves the cursor over the location on the map where the M4 Rifle was added on the mobile device, a pop up window is presented, as seen in Figure 40. The results listed in this pop up window are the TAMCN, NSN, serial number, time stamp of when the user added the item, and the description of the asset. Because the information presented in the figure matches the M4 Rifle with serial number 98765 found on the database, it proves that the location snapshot feature for the GCSS-MC mobile application is functioning properly.

Figure 40. Location Snapshot, Part 3

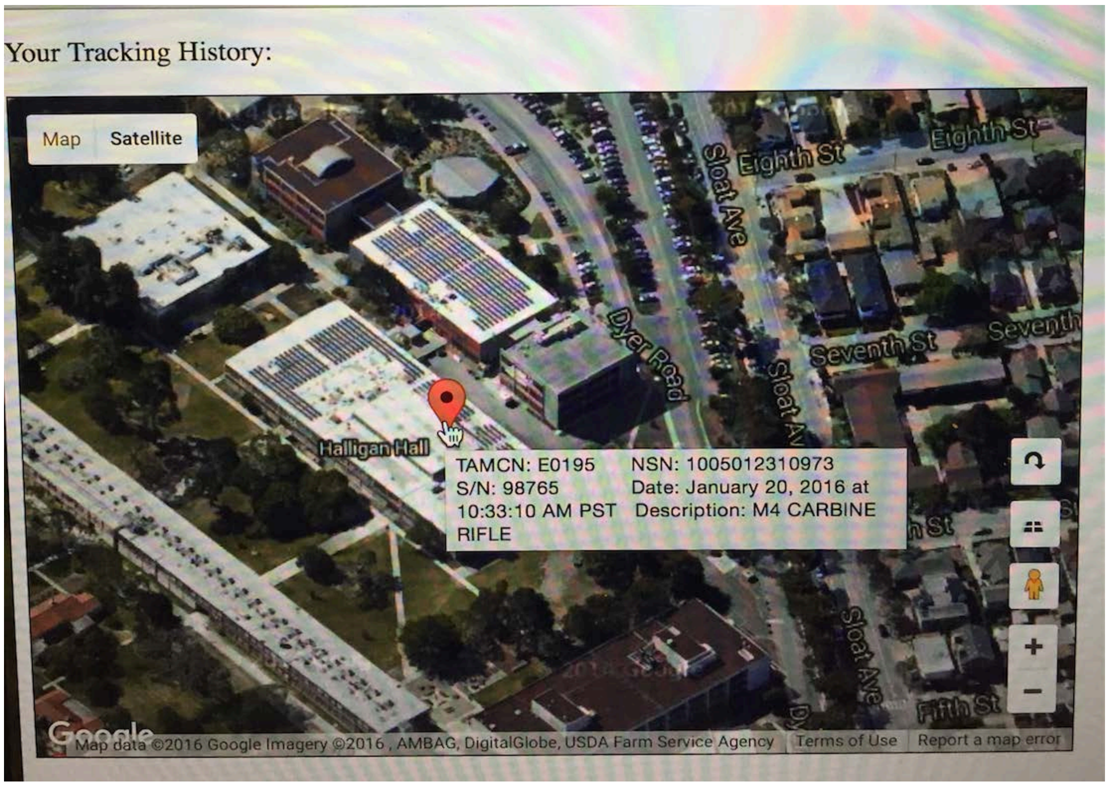


TAMCN: E0195 NSN: 1005012310973
S/N: 98765 Date: January 20, 2016 at
10:33:10 AM PST Description: M4 CARBINE
RIFLE

While the user may utilize the map format in Google Maps, there are other perspectives within this webpage that can be utilized. After clicking the “Satellite” button, which can be seen on the top left portion of Figure 40 and zooming in on the M4 Rifle that was recently added, the user will see a view similar to the one found in Figure 41.

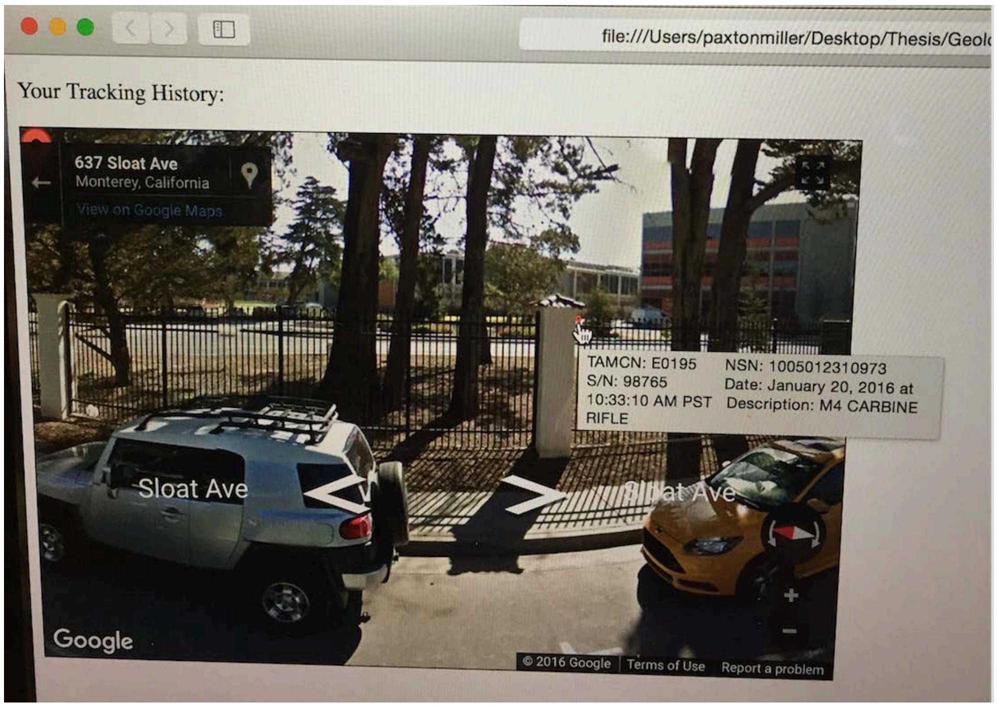
By selecting the street view button, the user will see yet another perspective. This provides a “first hand” view of where the user may find the location of assets in the system. Figure 42, in particular, shows a snapshot of where the M4 with serial number 98765 was added to the database. Like the map format of the location snapshot, each of these perspectives show pinpoints of where each asset should be located and a pop up window detailing specific data about that item.

Figure 41. Location Snapshot, Part 4


Your Tracking History:

TAMCN: E0195 NSN: 1005012310973
S/N: 98765 Date: January 20, 2016 at
10:33:10 AM PST Description: M4 CARBINE
RIFLE

Figure 42. Location Snapshot, Part 5


file:///Users/paxtonmiller/Desktop/Thesis/Geol

Your Tracking History:

637 Sloat Ave
Monterey, California
View on Google Maps

TAMCN: E0195
S/N: 98765
NSN: 1005012310973
Date: January 20, 2016 at 10:33:10 AM PST
Description: M4 CARBINE RIFLE

If utilized properly, having a feature like this can save units countless hours in locating assets. It will improve efficiency, property management, and allow a commander to have a full view of the items under his or her responsibility. Without the utility of a mobile device and the use of a mobile application like the one created for this thesis, this feature would be much more difficult to successfully employ.

D. SUMMARY

In Chapter IV multiples features were evaluated. These features were functions predominantly located in the property module of the GCSS-MC mobile applications, which included searching for an asset by NSN or serial number. Additionally, testing the app’s ability to add or delete an asset from the database, as well as capture the GPS coordinates of the mobile device, was also accomplished. Finally, a location snapshot webpage intended to map the coordinates of the assets in the database was evaluated. Chapter V provides the conclusions of this research and recommendations for future work.

V. CONCLUSIONS AND FUTURE WORK

Chapter V provides conclusions of the development and evaluation of the GCSS-MC mobile solution. It also suggests extensions to the current solution and areas of future research.

A. CONCLUSIONS

The user of mobile devices can support the way the DOD manages logistical requirements and dramatically improve productivity and efficiency across the supply chain. This thesis provides a proof-of-concept mobile application for GCSS-MC.

This work demonstrates scenarios, in Chapter IV, describing how accessing the GCSS-MC servers from a mobile device would enhance the user capability tremendously. This evaluation tests functions in the application that any user, such as commanders, supply personnel, or mechanics, could benefit from. The proposed solution not only creates a tailored suite of functions that would likely be utilized in the app, but showcases how mobile devices can enhance GCSS in more ways than just making refinements in usability and flexibility. This is seen in the application's ability to access the mobile devices' GPS sensor, which can improve property management accountability.

Based on the significant capability enhancements demonstrated in this thesis, the current GCSS Mobile Field Service plan being executed by the Marine Corps is recommended to have mobile devices included in the implementation.

Through the use of bridging technologies, like Cordova, the application could be developed utilizing HTML5. This tool eases the transition in extending GCSS-MC Mobile across multiple platforms, as part of the app's purpose would be to eventually allow all GCSS users, regardless of mobile OS, to have the ability to use their own personal device to perform administrative logistics functions.

B. RECOMMENDED IMPLEMENTATION EXTENSIONS

There are several implementation extensions that require further study in order to advance this application into a more viable mobile solution. These include the employment of a photo feature on the application, improving user permissions and access to data, and tying the GCSS-MC database to the application.

1. Photo Feature Implementation

Fully implementing the photo feature on the application is recommended. The benefits of such a function were discussed extensively in Chapter III and throughout the thesis. The way this could be explored is by sending photo files to the GCSS central servers or directly to a unit server that allows multimedia to be uploaded and downloaded. These photos could be sorted by metadata or user information for ease of organizing the photos. Having the photo feature assists all users, including supply personnel, mechanics, and asset account holders.

2. Login and User Permission Implementation

Currently the login page and user permissions are not developed or secure. Logging in with a user's personal password and allowing access only to locations in the database that user is authorized to is critical. Creating users, implementing various user permissions, and evaluating its effectiveness on the mobile application would be greatly valuable in further developing the GCSS-MC Mobile proof-of-concept application. Because security is such a critical requirement for any software the DOD uses, exploring how to assure the confidentiality, integrity, and availability of a mobile database application is certainly an area for future research.

3. Connecting to the GCSS-MC Database

The database created for this thesis is not fully representative of how the genuine tables and records appear on the actual GCSS-MC servers. Implementing the application to utilize an accurate representation of GCSS tables would be greatly beneficial. Not only would it provide a better indication of how the application performs with a database of a much larger scale, but it would also further identify if any data consistency issues were found in the current application implementation.

C. AREAS FOR FUTURE RESEARCH

While creating the application, areas for future research and development were found. Having an option intended for using the application in an austere network environment is a critical requirement if this project is to be employed. Additionally, genuine GCSS-MC user testing and evaluation of the application is recommended in order to objectively determine the overall effectiveness and efficiency of the system.

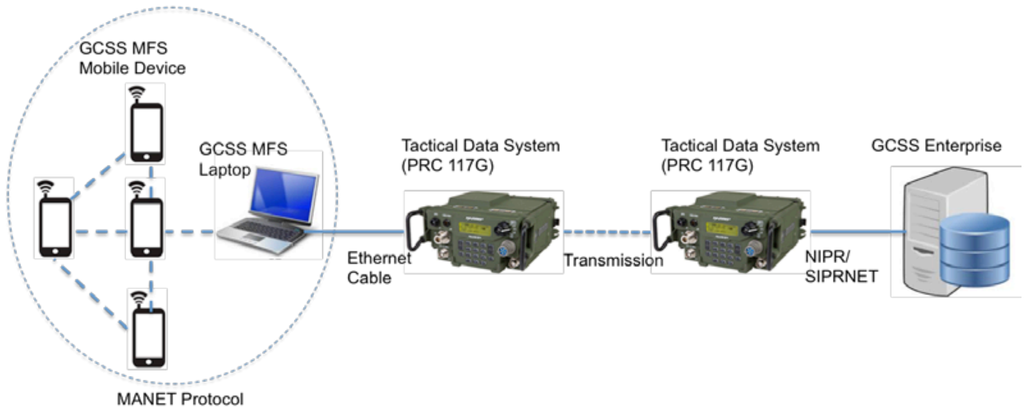
1. Austere Environment Option

One of the drawbacks to the current implementation of the application is the requirement to stay connected to the network in order to communicate with the database server. Because Marines often work in austere environments, the app needs a means to operate in these conditions. Fortunately, there are numerous options that can be explored to adjust to this shortfall. For mobile devices disconnected from the network, having the ability to perform operations in the same way the Marine Corps's current MFS plan is being implemented for laptops, as described in Chapter II, would be greatly beneficial.

Another area of future research is creating the ability for a user to download his or her unit's asset database or CMR account prior to disconnecting from the network. While this capability would not have a real-time snapshot from the GCSS-MC servers, it would still be greatly beneficial to the user. It would grant CMR users an easy-to-use means to view and make notes on the account for which they are responsible, give commanders an approximate idea of operational readiness, and allow supply personnel or mechanics a means to preload GCSS-MC commands or requests until a connection to the servers can be reestablished.

One of the inherent concerns with using assets that require electricity in an austere environment is the energy requirement that is essential for them to operate. Battery life on a laptop or mobile device can become a critical concern if generators cannot handle the energy they demand. Thus, reducing the power consumption that the application utilizes in this setting is another area for research in GCSS-MC Mobile. Maintaining awareness of the power or energy consumed on the actual device is significant, but power efficiency in how the mobile device sends packets through the network is a concern as well. Energy Aware Computing is often used on Mobile Ad-Hoc Networks (MANETs). MANETS are infrastructure-less networks that include, but are not limited to, mobile devices. These networks could be leveraged to communicate with an asset that is in fact still connected to the GCSS-MC servers. If a laptop is securely connected to the GCSS-MC server and a mobile device communicates with it through the means of a MANET protocol that takes energy consumption into account, then the laptop could serve as a conduit through which the mobile device could still access the GCSS-MC servers in an effective way. See Figure 43 for an example of a potential MFS architecture utilizing mobile devices.

Figure 43. MFS Architecture with Mobile Devices



Lastly, having the ability for a user to “turn on” an austere environment option manually would be beneficial, but exploring a “context aware” system would be even more advantageous. If the app was able to perform context-triggered actions, the switch to an energy-efficient austere mode could simply be a notification to the user. For example, if cellular or wireless networks lose a certain threshold of bandwidth, the app could automatically move to this option. Depending on how it is implemented, this anticipatory computation could mean the application takes a snapshot of the current unit database before losing signal completely or turns on power-aware functions to extend the life of the device. This would ultimately provide the user with an application that is more ubiquitous to the surrounding environment, reduces the workload on a bandwidth-constrained network, and further improves the efficiency of GCSS-MC mobile.

2. GCSS-MC User Evaluation

It is recommended that the GCSS-MC Mobile application not only have a standardized usability test produced for the prototype, but also utilize it to objectively assess the efficiency and effectiveness of the solution. One of the most imperative aspects of software implementation is testing and evaluating the application with authentic GCSS-MC users. There are number of Human-Computer Interaction (HCI) considerations that should be accounted for when determining the value, interest, usability, and effectiveness of the mobile solution. Gathering valuable test subject feedback and finding new areas for improvement will allow the mobile application to progress to an even more practical option for logistics support.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Apache Tomcat. (2015). Apache Tomcat. Accessed 11 December 2015. Available: tomcat.apache.org

Bitto, Nicholas. (2014). *Adding big data analytics to GCSS-MC* (master's thesis). Naval Postgraduate School, Monterey, CA. Available: http://calhoun.nps.edu/bitstream /handle/10945/43879/ 14Sep_Bitto_Nicholas.pdf?sequence=1

Cordova. (2015). Overview. Accessed 10 December 2015. Available: https://cordova.apache.org/docs/en/latest/guide/overview/

GAO 14-309. (2014). *Major automated information systems. Selected defense need to implement key acquisition practices*. Available: http://www.gao.gov/assets/670/662045.pdf

GCSS-MC/LCM Increment 1 CONOPS. (February 2014). Version 3.2. Quantico, Virginia: HQ USMC Installation and Logistics.

IT Dashboard. (2015a). Evaluation history. Global Combat Support System – Marine Corps Increment 1 Available: https://itdashboard.gov/investment/evaluation-history/59

IT Dashboard. (2015b). GCSS – Marine Corps Increment 1. Available: https://itdashboard.gov/investment?buscid=14

IT Dashboard. (2015c). Navy enterprise resource planning. Available: https://itdashboard.gov/investment?buscid=16

Jones, Mark. (2010). *Implementation challenges for DOD logistics enterprise resource planning IT systems.* Monterey, California: Naval Postgraduate School.

Kanaracus, Chris. (2012). Air Force scraps massive ERP project after racking up $1B in costs. *Computerworld*. Available: http://www.computerworld.com/article/2493041/it-careers/air-force-scraps-massive-erp-project-after-racking-up--1b-in-costs.html

Leach, Nile. (2006). Oracle Mobile Field Service User Guide. Release 12. Redwood City, CA: Oracle Corporation.

Leonard, Timothy and Philip Gallo. (2014). *Global Combat Support System-Marine Corps proof-of-concept for dashboard analytics* (master's thesis). Naval Postgraduate School, Monterey, CA. Available: http://calhoun.nps.edu/handle/10945/44567

Mobile Field Service Enhanced (MFS-E). (2009). Accessed 2 July 2014. Available: http://www.marcorsyscom.usmc.mil/sites/gcss-mc/index.aspx/fsmfs-e

Mutter, C. A. (1996). The Marine Corps Technical Publications System. Available: https://acc.dau.mil/adl/en-US/33745/file/6862/MCOP5215.17C %20The%20Marine %20Corps%20Technical%20Publications%20System.pdf

Oracle. (2014). What is SQL Developer? Accessed 11 December 2015. Available: http://www.oracle.com/technetwork/developer-tools/sql-developer/what-is- sqldev-093866.html

Oracle Mobile Field Service Data Sheet. (n.d.). Accessed 2 July 2015. Available: http://www.oracle.com/us/products/applications/056914.pdf.

PEOEIS. (2015). About GCSS-MC. Accessed 29 October 2015. Available: http://www.public.navy.mil/spawar/PEOEIS/GCSS-MC/Pages/About.aspx

Rouse, Margaret. (n.d.). ERP (enterprise resourse planning) definition. Accessed 18 November 2015. Available: http://searchsap.techtarget.com/definition/ERP

Supply CMRs and ROS. (2015). Accessed 29 October 2015. Available: http://www.hqmc.marines.mil/Agencies/HeadquartersandServiceBattalion/S4Logistics/SupplyOffice/CMRsandROs.aspx

Takai, Teresa. (2012). *Department of Defense mobile device strategy*. Washington, D.C.: Department of Defense.

Takai, Teresa. (2013). *Department of Defense commercial mobile device implementation plan*. Washington, D.C.: Department of Defense.

U.S. Marine Corps Concepts and Programs. (2015). Global Combat Support System – Marine Corps. Accessed 18 November 2015. Available: https://marinecorpsconceptsandprograms.com/programs/command-and-controlsituational-awareness-c2sa/global-combat-support-system-marine-corps

ViaSat. (2015). Secure mobile technology delivered to U.S. Marine Corps. Accessed 29 October 2015. Available: https://www.viasat.com/news/secure-mobile-device-technology-delivered-us-marine-corps

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California