Company Management System

Manual

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 26/03/2011 | 1.0 | Initial version | Michael Laws |
| 13/4/2011 | 1.1 | Add pages | Michael Laws |

Table of Contents

0. Customer’s Feedback 3

1. System Introduction 3

1.1 Purpose 3

1.2 General Overview 3

1.3 Customer and Technical Advisor Information 3

2. System Setup 3

2.1 Location 3

2.2 Administrative 3

3. High Level View of System 3

3.1 Django 3

3.2 The Company Management System 4

4. Quick Start Tutorial 4

4.1 Logging in 4

4.2 Submitting a MO 4

4.3 Updating personal information 4

5. Detailed View of System 4

5.1 Django 4

5.2 The Company Management System 4

5.2.1 MID 4

5.2.2 Bravo Inspection 6

5.2.3 Company Watch 7

5.2.4 Form1 8

5.2.5 Medical Chits 9

5.2.6 Accountability 10

5.2.7 Movement Orders 11

5.2.8 ORM chits 12

5.2.9 Special Request Chits 12

5.2.10 Uniform Inspections 13

5.2.11 Weekends 14

5.2.12 0800 Report 15

6. Index 16

7. Appendices 16

7.1 Troubleshooting/Known Bugs 16

7.2 Maintenance Procedures and Issues 16

7.3 Developers’ Information 16

Manual

# Customer’s Feedback

No feedback at the moment.

# System Introduction

## Purpose

The purpose of this document is to provide a detailed tutorial and description of the Company Management System (“cms” or “software” for short). This document describes the features within and stepwise instructions of how to use them. This document describes how our team sees and understands the workflow that the software is designed for and how to leverage that workflow for your use.

## General Overview

This application is a fully-functional company management web interface. It consists of many different modules that apply to all of the many moving parts that define managing a company. Some of the modules include: inspections, chits, and 0800 reports.

All paperwork that is converted to use in this system adheres to all Naval Academy standards set forth in official USNA orders.

This application is solely based on the web and accessible by anyone within the USNA network. The server handles all of the concurrency issues associated with the multiple users operating on the same database. The software on the server handles all the user authentication and data storage.

## Customer and Technical Advisor Information

|  |  |
| --- | --- |
| **Customer**  LT Igawa, USN  25 Company Officer  410-293-7525  [igawa@usna.edu](mailto:igawa@usna.edu) | **Technical Advisor**  Lori Delooze  Assistant Professor  410-293-6820  [delooze@usna.edu](mailto:delooze@usna.edu) |

# System Setup

## Location

The current URL of the cms is [www.usnacms.info/cms/](http://www.usnacms.info/cms/). This url is subject to change if this system is implemented. The entire system is a webpage and accessed by browsing to this url.

## Administrative

There is no overarching administrative functionality. Each user is given functionality based on their user level. The general user login will be the MIDN m”ALPHA#” format with a pre-selected password. Test accounts with limited (or relatively unlimited) access will be provided if asked for.

# High Level View of System

## Django

“[Django](http://www.djangoproject.com/) is a high-level Python Web framework that encourages rapid development and clean, pragmatic design.” There are three parts to the Django system: Models, Views, and Templates.

The benefits to using a web framework of this caliber and polish are numerous. The main ones include cleanly handled errors; much of the actual backend code is generated instead of laboriously typed; and the system is easily maintained and extended through use of applications. Please reference the detailed view below for additional detail. Django contains an administrative system that will be disabled for a deployment of the system. This system allows direct modification to the contents of the database.

## The Company Management System

CMS has a modular design consisting of different applications that perform different tasks. Applications can be used for any purpose, but most of them fill very specific roles. For example, the ‘form1’ application handles form1’s and is directly used through the web interface. The ‘mid’ application, on the other hand, just handles raw data that is used by all the other applications. Please reference the detailed view below for additional detail.

# Quick Start Tutorial

The fundamental paradigm here is that the user interface is supposed to be intuitive. If direct instruction is required, then the development team failed.

## Logging in

Reference section 5.2.1

## Submitting a MO

Reference section 5.2.1

## Updating personal information

Reference section 5.2.1

# Detailed View of System

## Django

As stated earlier, there are three parts to the Django system: Models, Views, and Templates. Models consist of the data models that are translated into raw MySql by the API. Views handle all of the website functionality. Templates are a simple text file that defines the look of the site and is treated similarly to any HTML file.

## The Company Management System

CMS itself is considered a project in Django syntax. As stated earlier, individual applications are used to define the modular design of the system. This implies a tiered application structure, but in reality it is very flat. Applications can include the data structures from any other application as python objects. Contained below is a detailed description of each application and its purpose, functionality, and use.

### MID

This application handles all of the attributes of a Midshipman. Every data point that defines a mid is contained within and is referenced by the other applications. The user is able to change applicable data within this module as well. This module also handles logging in and out, as that functionality is directly tied to the individual.



Login Screen. Input username and password.



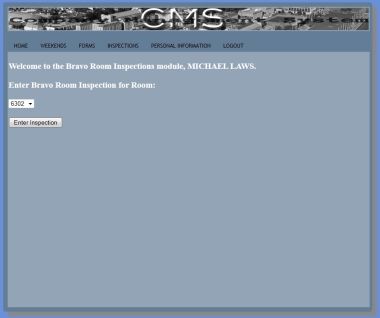
Modify personal information



Change password

### Bravo Inspection

This application handles entering and reviewing bravo room inspections. It is designed on a per room, per company basis. This is demonstrated in figure # and figure #.



Find a room to inspect



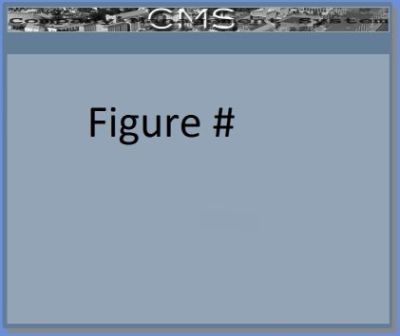
Uncheck the stuff that was hit.



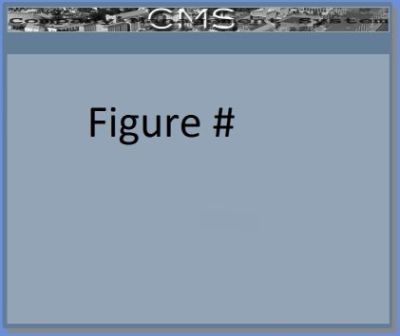
Review the who/what/when of the inspection

### Company Watch

This application provides functionality for the company to maintain a digital watchbill and logbook for CMOD, ACDO, and CDO.



Digital logbook



Watchbill

### Form1

This application handles the creation and storage of Form 1’s.



Review Form’s that your login has gotten



Input your own form1’s

### Medical Chits

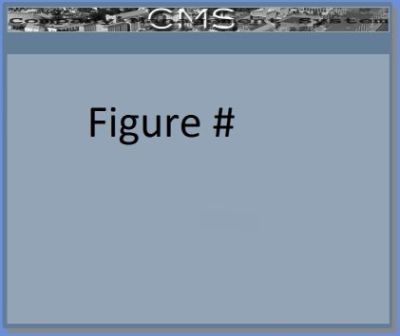
This application handles the creation and storage of Form 1’s.



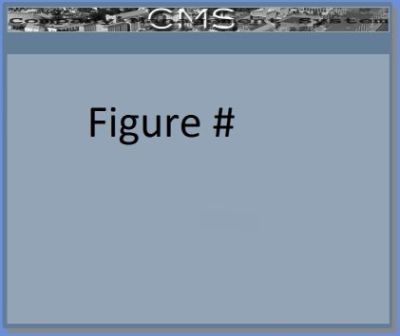
The well engineered medical chit page.

### Accountability

This application takes care of the accountability of Mids for formations, taps, football games, etc.



Enter accountability for a formation



Review accountability for an event.

### Movement Orders

This application allows the user to input and review Movement Orders.



Input a Movement Order



Review Movement orders.

### ORM chits

This application takes care of the ORM chit functionality.



Input and review

### Special Request Chits

This application takes care of the Special Request Chit functionality.



Input and review special request chits.

### Uniform Inspections

This application takes care of the Uniform inspection functionality.



Input a uniform inspection.



Review a uniform inspection

### Weekends

This application takes care of requesting, approving, and listing weekends.

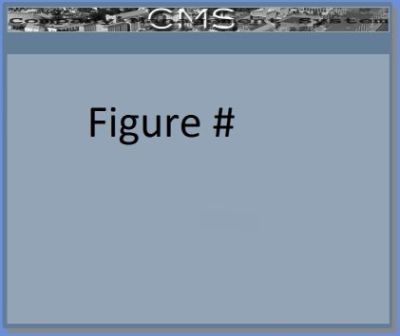


Request a weekend. Past the deadline, so unable to request.

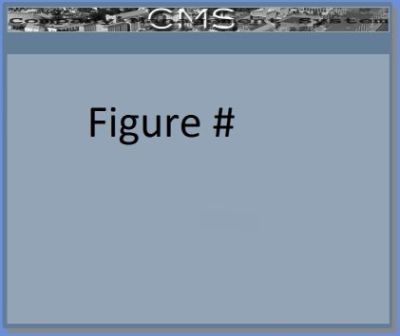


Reviewing an empty weekend list.

### 0800 Report



Entering data into a 0800



Reviewing past 0800 reports.

# Index

As previously stated, this system is supposed to be intuitive; therefore an index would be an unnecessary writing endeavor.

# Appendices

## Troubleshooting/Known Bugs

None at this time

## Maintenance Procedures and Issues

None at this time.

## Developers’ Information

|  |  |  |  |
| --- | --- | --- | --- |
| Michael Laws  Team Lead  [michaellaws@gmail.com](mailto:michaellaws@gmail.com) | Michael Harrison  Models/Database  [marinemike1985@gmail.com](mailto:marinemike1985@gmail.com) | Dimitri Hatley  Views/site functionality  [hatleyd@gmail.com](mailto:hatleyd@gmail.com) | Ryan Rabe  Templates/CSS/AJAX  [ryanrabe2011@gmail.com](mailto:ryanrabe2011@gmail.com) |