Integral Solutions

University of Maine

COS 497 | Capstone II

Administrator Manual

Digital Program of Study Approval System (POSAS)

Client: Doctor Harlan Onsrud

Team Members:
Vincent King | Peter Riehl | Mac Creamer
Liam Blair | Aaron Wilde

Professor:
Terry Yoo

April 5, 2022

Version 1

Digital Program of Study Approval System
Administrator Manual (AM)

Table of Contents

1. Introduction

1.1 Purpose of This Document

This document serves as an administrative manual for the future developers and maintainers of the *Digital Program of Study Approval System*, initially developed by Integral Solutions as a Computer Science Capstone project at the University of Maine for our client Harlan Onsrud. The intended readership of this document is the current and future system administrators who are tasked with the continued maintenance and development of this project, the *Digital Program of Study Approval System.* This document goes over the past references and deliverables of this project including our initial proposal, our System Requirements Specification (SRS), our System Design Document (SDD), our User Interface Design Document (UIDD), Our Critical Design Review Document (CDRD), and our Code Inspection Report (CIR). This administrative manual goes over the background of our project from an administrative perspective and includes details about our systems software and hardware components and requirements. Administrative procedures such as installation, routine tasks, periodic administration, user support, and known bugs will be covered in this document for future administrators to gain full understanding of our system and its components and limitations.

References

- "UMaine Graduate Student Program of Study Creation and Approval System" Proposal
    - *Author: Harlan Onsrud*
    - *Date: September 2021*
- "UMaine Graduate Student Program of Study Creation and Approval System" SRS
    - *Author: Integral Solutions*
    - *Date: October 2021*
- "UMaine Graduate Student Program of Study Creation and Approval System" SDD
    - *Author: Integral Solutions*
    - *Date: November 2021*
- "UMaine Graduate Student Program of Study Creation and Approval System" UIDD
    - *Author: Integral Solutions*
    - *Date: November 28, 2021*
- "UMaine Graduate Student Program of Study Creation and Approval System" CDRD
    - *Author: Integral Solutions*
    - *Date: December 2021*
- "UMaine Graduate Student Program of Study Creation and Approval System" CIR
    - *Author: Integral Solutions*
    - *Date: March 2022*

2. System Overview

2.1 Background

With regard to the software side of POSAS, the system uses Django as both a frontend and backend. This means that POSAS utilizes the native Django Administrative panel, available at '<host domain name>/admin' (e.g., 10.0.0.1/admin). At this moment in time, the control that is had over the system within the administrator position involves accessing account information and the ability to change a user's email, password, and other affiliated account fields. As the system continues to develop, being able to access, correct, delete, and modify POS's will become an element of the system, as will being able to update associated fields. The reasoning behind this is, though this is a lot of power on the administrative side of things, if a workaround for an unforeseen bug is needed, it will be helpful to have one on hand. Account information for the administrative accounts will be not included within this document, but will be provided to the administrator of the system as the time comes necessary to release said information. Additionally, the system administrator has access that any given student/staff has to review a POS, and the same capabilities that a student has to create a POS, to ensure that nothing regarding the process is broken.

With regard to the hardware side of POSAS, the administrator has access to the virtual machine from the Advanced Computing Group (ACG) that hosts the system. The administrator will have the ability to edit and maintain the docker container that houses POSAS, while also being able to contact ACG for any further requests or information necessary to continue POSAS operations. Due to the fact that this system is hosted by ACG, no discussion regarding the hardware-based setup is necessary, as spinning up the container is possible via the command line interface that is accessible through an ACG account connected to POSAS. The database is housed in a separate docker container than the one that houses POSAS, so should the website itself crash, no information is harmed and another container can be instantly spun back up to continue operations as though nothing happened.

2.2 Hardware and Software Requirements

Hardware requirements, due to the system being based within ACG's hosting servers on-campus, are limited. Currently, we utilize little in the way of RAM and storage for the server, however, should the entire graduate school eventually become integrated with this system, than more RAM and storage space may be needed. A rule of thumb you can go by with regard to server hosting is that for every 50 concurrent active users you have, 1 GB of RAM is needed for smooth operations. It is unlikely that the system will have frequent visitors throughout the year, so planning for 4 GB of hosting RAM would be more than sufficient. As for storage space, each user has several profile fields attached to their account, and anywhere between 1 and 5 programs of study on average. If we assume the average-case user has 3 programs of study, and that each program of study (and all of its approved versions) has to be stored away, then we can roughly assume for each user, they will need 2 MB of storage. This is likely a high estimate, however, this gives a safe frame within which to work. With 4 GB of storage, the system would be able to house roughly 2,000 students. Far greater amounts of storage can be allocated, with the ability to add 500 more students to the system with each additional GB of storage added.

With regards to software requirements, many of the packages and libraries necessary to run POSAS with Django are able to be automatically downloaded and installed through a command line interface. Such a process is described below in section 3.1. The packages necessary at this moment in time to run POSAS, however, are as follows:

- asgiref
- aiohttp
- async-timeout
- attrs
- chardet
- Django
- idna
- multidict
- psycopg2
- python-decouple
- pytz
- sqlparse
- typing-extensions
- websockets
- yarl

Each of the above required files is possible to be installed through python's `pip install` command, but for the purposes of the document, an easier methodology is described in section 3.1. Additionally, installing `git` within your system will make version control and access significantly easier in the long run. As all the code is currently hosted within GitHub, creating and accessing the repository there via git will ultimately result in more successful maintenance in the future.

3. Administrative Procedures

[Note: These sections are required. However, you may have other information that a system administrator must be aware of. If so, simply add more sections beginning with Section 3.5.]

3.1 Installation

The majority of the installation tasks are handled using docker-compose. Docker uses a dockerfile that outlines everything required to build our application and then builds each container. The system has two containers, one for an nginx web server and a separate one for the PostgreSQL database. The administrator will have access to the dockerfile on the virtual machine. This file contains secret information such as the database password, the port to access the database, and system information about the virtual machine.

The administrator is not required to do many installation tasks. The virtual machine contains all of our source code and is already set up to host our application. Specifically, it is already containerized user Docker with our database and web server. As long as the virtual machine is online, the website and database should be accessible.

3.2 Routine Tasks

Routine tasks necessary to be undertaken are limited in nature due to the mostly self-sufficient nature of POSAS. Once in a while, assuming that a user reaches out for support with the service, the administrator will have to respond to that user and assist them with their problem. Recreating that problem locally, if applicable, will be necessary for the purposes of successfully walking the user through the issues they are having. Additionally, creating accounts for staff members will also be a necessary routine task that has to be performed. Staff members will have their accounts created for them, to avoid any potential fraud in attempting to do the account creation system in an automated manner for the staff. This “real-person” verification system that involves emailing, zooming, meeting in-person, or whatever modality of verification is preferred, will prevent abuse cases where a student attempts to create a staff account.

One other task that may have to be completed occasionally is monitoring of storage space relative to the number of users who are within the system. Ensuring the prevention of abuse where a user is spam creating as many POS’s as they can will be preferred to maintain longevity of the system.

3.3 Periodic Administration

Our system will be mostly hands off. People who are performing period tasks will have to keep track of the following things: supporting software updates, UMaine Advanced Computing Group (website hosts) updates, and user account issues. Each of these will be fleshed out below.

Our system uses a variety of software to accomplish the POSAS website. These softwares include HTML, CSS, PostgreSQL, Python, and Django. Each of these softwares, when the web service was created, was running the most updated version of itself. As time goes on new versions of each software will be released by the developers and our system will possibly need periodic updates. While each new release of the individual softwares may not be critical to the functionality of the system, updates once a year must be mandatory so the system does not fall too far behind on updates. Update procedures for all these software are easily accessible on developers websites as well as tech help websites. A quick google search should suffice.

Our system is hosted from the UMaine Advanced Computing Groups (ACG) servers here on campus. Occasionally the ACG will announce that they are running updates to their machines or that administrative action must be taken on users websites. The administrator of the website should be named the primary owner of the server space provided by ACG. These periodic updates will not always require action from the administrator but should be noted and an email should be sent out prior to the scheduled server update times letting users know that periodic outages may be experienced.

Lastly, our website may need upkeep with the user accounts. While there is a feature integrated into the system that deletes accounts who are identified with an internal algorithm as being inactive for 2+ years, there may be times where the administrator may be called upon to manually delete accounts. Since our website all handles passwords and some user information it may be required of the administrator to help reset passwords. Our system has an integrated reset password function that should handle these issues but the administrator may be contacted through the website email for more in-depth help.

3.4 User Support

Due to our system being so heavily integrated with user accounts our website should be, and is, integrated with some support features. Contact information will be provided on the website's main page that will direct users to a support email and the administrators email so that account and software issues can be discussed or passed along for documentation and repair. Online support such as technical support, calling / live chat, will not be available to users. Help pages and text boxes in our website's user interface will hopefully relieve the majority of the issues that users encounter as they utilize our web service. As stated above, for more in depth questions or concerns, our system administrator will be linked to the website and can be contacted by users for front end support or backend questions.

4. Troubleshooting

4.1 Dealing with Error Messages and Failures

First and foremost, once the system is in the production environment, the debugging mode will need to be disabled. Minor issues the site will experience suddenly turn into full-page debug error messages. The debugging mode should only ever be turned on within local environments that mimic the live environment, as opposed to the live environment itself.

In the event that a serious error message or failure occurs, attempt to reproduce the issue within a local environment that mimics the live environment. Within this setting, enable debugging so that error messages are displayed. Though it may seem obvious to some, or questionable to others, when you are able to reproduce the error within a local environment, begin searching the web to see if other developers or administrators have experienced a similar issue. The chances are high that you will not be the first to experience the error, and you likely won’t be the last. If the issue is so serious that the live environment is completely down, immediately restart the docker container and try to spin the environment back up again. Restarting and spinning the docker container up again effectively gives a clean slate to the front end of the site. This allows any issues on this end to be resolved without fundamentally impacting the contents of the site itself.

It is not recommended to go scouring through the code when you encounter an error, despite the information provided by the debugging being enabled, unless you have significant experience in doing so. For many, this ends up being a wild goose chase that yields no end results, while for a few, it has the chance to show what element of the code is causing the issues at hand. Regardless, the first thing that should be attempted is to search online for the error message being displayed.

4.2 Known Bugs and Limitations – Aaron

As far as limitations go, they are abundant at the time of writing. Some files, such as those used for graduate form information input, are lacking certain fields, and do not extend the base file when required. These defects are still present in the current state of the system purely due to time constraints, and at the time of writing, are planned to be ironed out soon.

Other bugs present in the system are discussed in Integral Solutions' Code Inspection Review document; such defects include those in logic errors, coding conventions, security oversights, user friendliness, and more. These have not been fixed yet, and at the time of writing, only the first two can be realistically expected to be ironed out in time for the release. The other problems will most likely be present in the first build purely due to the complexity of said problems.

Appendix A – Team Review Sign-off

By signing below, both parties confirm that they have reviewed the contents of this document. Additionally, both parties will confirm that they have agreed on the document's content and format.

Team Member Comments:

1. ____
____
____
2. ____
____
____
3. ____
____
____
4. ____
____
____
5. ____
____
____

Customer Name: ___________________ Customer Signature: ___________________

Date of Signature: ___________________

Team Names:	Team Signatures:	Date of Signatures:
___________________	___________________	___________________
___________________	___________________	___________________
___________________	___________________	___________________
___________________	___________________	___________________
___________________	___________________	___________________

Appendix B – Document Contributions

Liam Blair - 10%

- 3.1 - Installation

Mac Creamer - 35%

- 4.1 - Dealing with Error Messages
- 3.2 - Routine Tasks
- 2 - System Overview

Vincent King - 15%

- 1 - Introduction

Peter Riehl - 30%

- 3.3 - Periodic administration
- 3.4 - User Support

Aaron Wilde - 10%

- 4.2 - Known Bugs and Limitations

All

-