# CMSC 345

### Software Design and Development

# Piece of Eden Administrator Manual

Piece of Eden Rentals

Administrator Manual

#### Table of Contents

Page

## 1. Introduction

1.1 Purpose of This Document

1.2 References

2. System Overview

2.1 Background

2.2 Hardware and Software Requirements

3. Administrative Procedures

3.1 Installation

3.2 Routine Tasks

3.3 Backups

3.4 User Support

4. Troubleshooting

4.2 Dealing with Error Messages and Failures

4.3 Known Bugs and Limitations

Appendix A – Peer Review Sign-off

Appendix B – Document Contributions

## 1. Introduction

1.1 Purpose of This Document

The purpose of this document is to familiarize a system administration with the running and maintenance of Piece of Eden Rentals

* 1. References
* System Requirements Specification
* User Interface Design Document
* System Design Document
* README.txt located in github “Code” directory

2. **System Overview**

2.1 Background

A High level design overview shows several major aspects of the software. Figure 1.1 in the System Design Documents illustrates the layout. At its heart is Django, which runs the python web framework central to the software. Another major aspect is MongoDB which runs the database. The Django ToolBox was installed to assist in linking Mongo and Django, and the PyMongo package is present to augment Mongo. Also in the backend, skel-js handles the CSS generation and Google Calendar API and Maps API plug into the web front-end. Originally it was planned that the web server should run off of AWS, but in the end the requirement was lifted and we run it on a private virtual web server.

Installation and Day-to-day mainenance is simple. It includes installation (detailed below), and making sure the virtual environment and web server are running. Modifying user accounts can be done from the admin panel (127.0.0.1:8000/admin).

2.2 Hardware and Software Requirements

The software requires very little hardware, since it is run off of a virtual webserver it can be run on any platform in a browser.

3. **Administrative Procedures**

3.1 Installation

To install, instructions must be followed from the README.txt in the github folder “Code.” It is also listed here for reference:

PART A: Installing Virtual Environment

Note: Commands run from command line are prefeaced with $

NEW INSTRUCTIONS:

1. Install python 2.7

Check if already installed:

$ python --version

Install:

Windows:

<https://www.python.org/downloads/>

Linux

#apt-get install python

2. Install pip if not already installed (very likily that it is)

Update it just in case it is not up to date

$ python -m pip install --upgrade pip

If not already Installed

Linux:

# apt-get install python-pip

3a. Install python virtual enviroment:

$ pip install virtualenv

3b. Set up virtual environment

To create folder venv (or name of your choosing) containing python executables and copy of pip library to install other packages

$ virtualenv venv

Reference: <http://docs.python-guide.org/en/latest/dev/virtualenvs/>

NOTES: It is a good idea to install this in an easy to get to place. You will will need to go to the directory and activate the virtual environment everytime you want to run the application.

Suggest making the directory and then clone the git repo again inside of that directory.

The virtual enviroment does not modify system files or configurations, it is just a standalone python install to a directory. Copy and paste completely moves it and deleteing the directory removes it.)

The virtual enviroment installs python in a local directory. This way we can keep all of the extra packages (django, mangoDB, South) we need in a single directory. At the end we can zip the directory and upload it to our repo as a deployable product.

3c. Install libraries (linux only)

# apt-get build-dep python-imaging

# apt-get install libjpeg8 libjpeg62-dev libfreetype6 libfreetype6-dev

4. Activate your virtual enviroment

windows:

$ Scripts\activate

linux:

$ source venv/bin/activate

Packages

5. DJANGO.

In order to get Django to work with mongo we need to install a special version of Django that works with no-rel database structures like mongo. To do this we need to install Django from source.

5a. Go to this git repo:

<https://github.com/django-nonrel/django>

5b. Click on the "Download ZIP" button to download a zip of the repo

5c. Unzip the folder in your virtual enviroment

5d. From within your ACTIVATED virtual enviroment run the following commands:

$ python setup.py build

$ python setup.py install

6. Install MangoDB engine for django

From ACTIVATED virtual environment:

6a. Install Djangotoolbox

$pip install git+<https://github.com/django-nonrel/djangotoolbox>

6b. Install Django MongoDB Engine

$pip install git+<https://github.com/django-nonrel/mongodb-engine>

Reference: <http://django-mongodb-engine.readthedocs.org/en/latest/topics/setup.html>

Note: DATABASE IS ALREADY CONFIGURED so no need to worry about that last step in the full instructions.

7. Install South

$ pip install south

8. Check packages

From ACTIVATED virtual environment

$ pip freeze

to list all installed packagesYou should see this:

Django==1.6.11

django-mongodb-engine==0.6.0

djangotoolbox==1.8.0

pymongo==3.2.2

South==1.0.2

Install Complete

PART B: Running Virtual Environment

You should have completed Part A at this point.

To Run the Server:

1. Enter virtual environment:

windows:

$ Scripts\activate

linux:

$ source venv/bin/activate

2. Change directory:

Section-1-Team-3/Code/mysite

3. Start Server

$ python manage.py runserver

The server is now running, use CNTL-C to kill it

Visit:

<http://127.0.0.1:8000/polls/>

<http://127.0.0.1:8000/beach_homepage/index.html>

<http://127.0.0.1:8000/accounts/login/>

<http://127.0.0.1:8000/admin>

PART C: Notes

Runserver command starts the dev server on the internal IP at port 8000

Now we have a deployable enviroment that we can put anywhere, as long as we do all the configuration in the pyhon virtual enviroment.

If you add any packages, check which packages with $ pip freeze and please add them to the list in step 8.

3.2 Routine Tasks

There are just a couple routing tasks.

1. Starting the virtual webserver. Once the system is installed, you must start the webserver by navigating to the virtual folder and running the commands:

$ source bin/activate

To activate the webserver, then change directories to:

(github repo location)/Code/mysite

Then run the command:

$ python manage.py runserver

2. Adding admin accounts can be done by running the command:

$ python manage.py createsuperuser

1. Accessing the admin panel allows you to add/delete/modify users. It is available at:

<http://127.0.0.1:8000/admin>

3.3 Periodic Administration

Account cleanup is entirely optional, but can be done in the admin panel referenced above.

3.4 User Support

User support will be given if someone loses their password or account name. Currently, there is no account reclamation tool for users to use. Also, disputes between users must be handled by a person and over email.

4. **Troubleshooting**

4.2 Dealing with Error Messages and Failures

Error messages and failures may occur when first installing the software. If you are setting up a new system, please refer to the instructions above on installing the virtual environment and web server.

4.3 Known Bugs and Limitations

The current set-up of having the website run on a virtual server is obviously a huge limitation. The fact that you can not access this website over the internet is a problem of not being able to host on AWS as was planned in the beginning of the project. There is a bug involving first and last names as well as emails not being stored to the user account database.

**Appendix A – Team Review Sign-off**

All team members have reviewed this document and agree both on the content and the format. Any concerns are addressed in the comment section below.

**Team** (print name in first blank, signature in second blank)

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Appendix B – Document Contributions**

Nicholas Keckeisen was the primary creator of this document, but information was contributed by all team members to add to it. For example, the README.txt document which was included within this document was originally created by Eric Forte, but has been heavily edited throughout the project.