SIMCON
ARC Deployment Guide

Installation:

1. Copy the simcon.tar file to the host directory for the website files.
2. Decompress the simcon.tar file
    a. Terminal Interface
        i. If you are not in the host directory, change directory to the host directory.
        ii. Execute the decompress command in the terminal (tar –xf simcon.tar)

Setup:
In order to deploy the Django Simcon application, the settings file will need to be configured for your hosting environment. Below are the minimum changes to the settings.py file that are required to deploy the application.

I. Update the following in the simcon/settings.py
II. Ensure that the DEBUG & TEMPLATE_DEBUG settings are set to False
III. Add import pytz
IV. Update TIME_ZONE = 'PST8PDT
V. Update the ALLOWED_HOST setting to include the website host.
VI. Update the MEDIA_ROOT setting to point to the location where the media files will be hosted from (including the recorded conversations).
VII. Update Email Settings (as needed)
    a. Update the EMAIL_HOST setting
    b. Update the EMAIL_PORT setting
    c. Update the EMAIL_USER setting
    d. Update the EMAIL_HOST_PASSWORD setting
    e. Update the EMAIL_USE_TLS setting as needed
    f. Update the DEFAULT_FROM_EMAIL setting
VIII. Update Database settings
    a. By default, Django use SQLite for the database server. No changes needed if you plan to use SQLite. For all other database servers, please see Django's documentations for configuration instructions (https://docs.djangoproject.com/en/1.6/ref/settings/#databases)
    b. Using ARC's MySQL server for the database server set DATABASE equal to the following

```
Database={'default': {
                    'ENGINE': 'django.db.backends.mysql',
                    'NAME' : 'database',
                    'USER' : 'username',
                    'PASSWORD' : 'password',
                    'HOST' : 'host',
                    'PORT' : 'port',
                }
            }
```

3. If a virtualenv has not been setup in the same directory as your manage.py file, setup a virtualenv as follows. (Note you must use a bash shell for the following steps).
    a. Connect to the ARC server using ssh terminal
    b. cd /path/to/dir/with/manage.py/in/it
    c. virtualenv –no-site-packages .env
    d. source .env/bin/activate
    e. Install required python packages
        i. pip install django
        ii. pip install MySQL-python
        iii. pip install pytz
4. Add the following to the wsgi.py file

```
import site
import sys

prev_sys_path = list(sys.path)
root = os.path.normpath(os.path.join(os.path.dirname(__file__), "../"))
sys.path.append(root)
site.addsitedir(os.path.join(root, ".env/lib/python2.6/site-packages"))

# addsitedir adds its directories at the end, but we want our local stuff
# to take precedence over system-installed packages.
# See http://code.google.com/p/modwsgi/issues/detail?id=112
new_sys_path = []
for item in list(sys.path):
      if item not in prev_sys_path:
      new_sys_path.append(item)
      sys.path.remove(item)
      sys.path[:0] = new_sys_path
```

5. Create the database.
    a. Terminal Interface
        i. If you are not in the host directory, change directory to the host directory.
        ii. Execute the sync database command in the terminal
            1. python manage.py syncdb
6. Collect the static files
    a. Terminal Interface
        i. If you are not in the host directory, change directory to the host directory.
        ii. Execute the sync database command in the terminal
            1. python manage.py collectstatic
7. Notify ARC of the path to path to the wsgi file, static files, and media files are to be served from. They will create an Apache VirtualHost for your project, and deploy it on our web server.

Updating the site:

1. Copy/Update the files that need to be updated.
2. Connect to the server via SSH
3. From a bash shell enter: touch /path_the_wsgi.py/wsgi.py