Capstone: Operating Instructions

Austin Cooperrider

Trident University International

CSC 425: BSCS Integrated Project

Mr. Michael Greco

29 Sep 22

# How to start

After downloading all files, open a windows command prompt and change directory to the folder that contains ‘main.py’. From there run the following command ‘python main.py’. This will launch the program and provide you the link to access the website. To access the website, just enter the link provided by the application into a web browser. The default link should be <http://127.0.0.1:5000> . Once you gain access to the website, all remaining functions will be through the web user interface.

## Key requirement

The most current version of python is required to run the program. If you do not have a valid version, it can be downloaded via their website <https://www.python.org/downloads/>

click==8.1.3

colorama==0.4.5

Flask==2.2.2

Flask-Login==0.6.2

Flask-SQLAlchemy==2.5.1

greenlet==1.1.3

itsdangerous==2.1.2

Jinja2==3.1.2

MarkupSafe==2.1.1

SQLAlchemy==1.4.41

Werkzeug==2.2.2

## Key Application Functions

The first function will be seen in the login capability and the fact the program can identify what type of user is logged in. a.admin and u.user are an admin and standard user. The user account does not have easy links to some pages because they do not have elevated privileges. Those pages will have further safeguards to ensure that even if the user got to those pages they cant commit any edits. The most obvious item is a user can not close an open ticket, where an admin can.  
 The most important function is the ticket creation and editing. The home page displays all of the tickets with some details about them. The buttons next to those tickets are links to their specific html link which preloads all data in the ticket page. The associated ticket page identifies if you are editing a ticket or creating a new one. This prevents creating multiple tickets for the same problem. Entries for the tickets will be displayed and saved on this page, thus allowing seamless updates.

The last key function is the adding and editing of users to the website. The admin page will have links to the add user page along with similar capabilities to the home page and ticket page relationship. So an admin can initiate a new user addition or edit a user status with some safeguards in place.

## Proof of concept

The project currently is showing the login function completely. The only additions needed are small scale user account validation at important updates and changes. Otherwise, it is fully functional.

The ticket display and editing is fully functional. Some editing associated with how they are displayed is to be expected but right now we are at the stage of functionality complete. The entries has not been tackled yet but should be relatively simple. As we have already proven the capability of displaying specific tickets information on the ticket page, it should not be a challenge to continue that with entries.

Similarly, the user listing and associated change page is almost a mirror of the home and ticket page. From there it is just duplicating the effort with those associated variables.

## Status

For user account functionality association to elevated position or not, it is almost complete. Going through every function that an admin should only be able to do, i just need to ensure we check if that status is there.

For the home page, I am currently displaying the tickets as a list item. There is probably some better display function i will need to research. From there, it should be simple to implement. The ticket page probably has the most functionality i need to work on. The key obviously being the creation of note entries in the database and the associated displaying of them. From there, how those entries will be displayed. Additional display item problems is the fact that the dates only display the date and not the time. That will have to be investigated too. The biggest challenge in this section was how to preload the information for specific tickets. Utilizing a subroute along with the default route allowed that to become a passed through variable. It also makes it a feature of if you know the ticket number you need to edit, you can just hand jam it into the url.

The remaining admin pages and their functionality needs to be worked. At this time, all they are doing is displaying the current info. The next step would be to edit the pages to actually display similarly to the ticket page and then allow database edits from there.

Overall, all of the remaining items are reasonably doable and just will take time to go through, code every item, and debug any typo.