

communication

inbound methods from which the user can communicate with the application and register support-issues.

facebook
twitter
irc
skype
sms
e-mail

<<core.py>>

Main system component

Initializes everything, the up-kicker of all the different components and services that is vital for the system to run and also loads, backups and restored the database when needed.

queue
communication
backend
variables (settings)

Queue

The queue module keeps track of all the active and non-active cases that has been reported and will be reported. There's no real way of fiddling (modifying) the active queue without going through the proper functions. All functions of the queue instance will report back and create a safe environment for the cases.

add()
accept(<id>)
complete(<id>)
queuetime(<category>)
notify() - Discontinued
_get()
get_queuepos()

Backend

Listens and creates a way for administrators to communicate with the core, the communications platform is for the users and their way of getting "through" to the core while the backend is the place where administrators can go in and modify queues, tweak settings and just get a overview of the system.

backend socket
api that works well with everything

<<backend.py>>

Backend GUI

A OpenGL based GUI which uses the api from the backend to get a graphical overview of the current situation within the core. This GUI can be replaced with a webinterface or anything that can interperate json data, but the GUI is neat and was fun to create.

OpenGL