

# **DASA Web Control System (WCS)**

## **System Design**

### **Introduction**

The DASA Web Control System (WCS) is a web application designed to be used as a tool to manage an automated showering system called DASA. DASA is a system which manages the showering head, temperature, and shower mode of a shower through voice commands or remote control via the WCS. The WCS's primary distinguishing features include:

- Logging shower statistics
- Managing shower profiles
- Controlling the shower remotely
- Altering settings of the DASA system

All these capabilities must be included in the system design.

### **Design Goals**

The WCS will be released as a fully functional tool, and must work seamlessly with the DASA shower system. The design of the WCS must always keep in mind its task at hand, and must be built in concurrence with the DASA shower system.

On top of this, the design must always keep in mind the aspect of safety. Since DASA is intended to be used by handicapped individuals, the control of the shower system on the web side must always maintain safety above all else. The WCS must communicate with the

showering system at a level that has very little lag, no locking, and no repetition or corruption of data.

Security is also a driving factor. Since the WCS will technically be accessible on a local network, it must not contain holes in which an uninvited agent could intercept, change, or corrupt data in any way. Security is also important in regards to credentials. The WCS will have two forms of authentication: the overarching shower account, which is for the entire system as a whole and is the frontier at the network level, and then the individual profiles, which are the accounts within the system that manage the separate instances of a shower. Obviously the shower account must be the most secure; it should not allow unknown or uninvited users to create new profiles or access the system. However, the individual profiles need to have a certain level of integrity as well; the corruption of a profile from another user would be unacceptable.

### **Key Components**

Six key components define the overall structure of the WCS system, and each provide a crucial role in implementing the features:

- User Interface
- Log Manager
- Profile Manager
- Database (for profiles & logs)
- Shower Manager (for communication with the shower)
- Emergency Manager

The user interface manages the visuals displayed for the user (obviously) but more importantly handles the representation of the shower controls (sliders for temperature, interactive image for head position, etc.).

The log manager handles anything involving the creation, access, or interpretation of logs within the system. It manages how logs are structured, as well as how they get passed from component to component.

The profile manager deals with the authentication, creation, deletion, and usage of profiles. The term “usage” in this instance is in regards to how profiles are used within the system; as the controllers of the shower itself. Therefore, the profile manager handles the initial creation and passage of commands to the shower, but does not actually pass them to the shower.

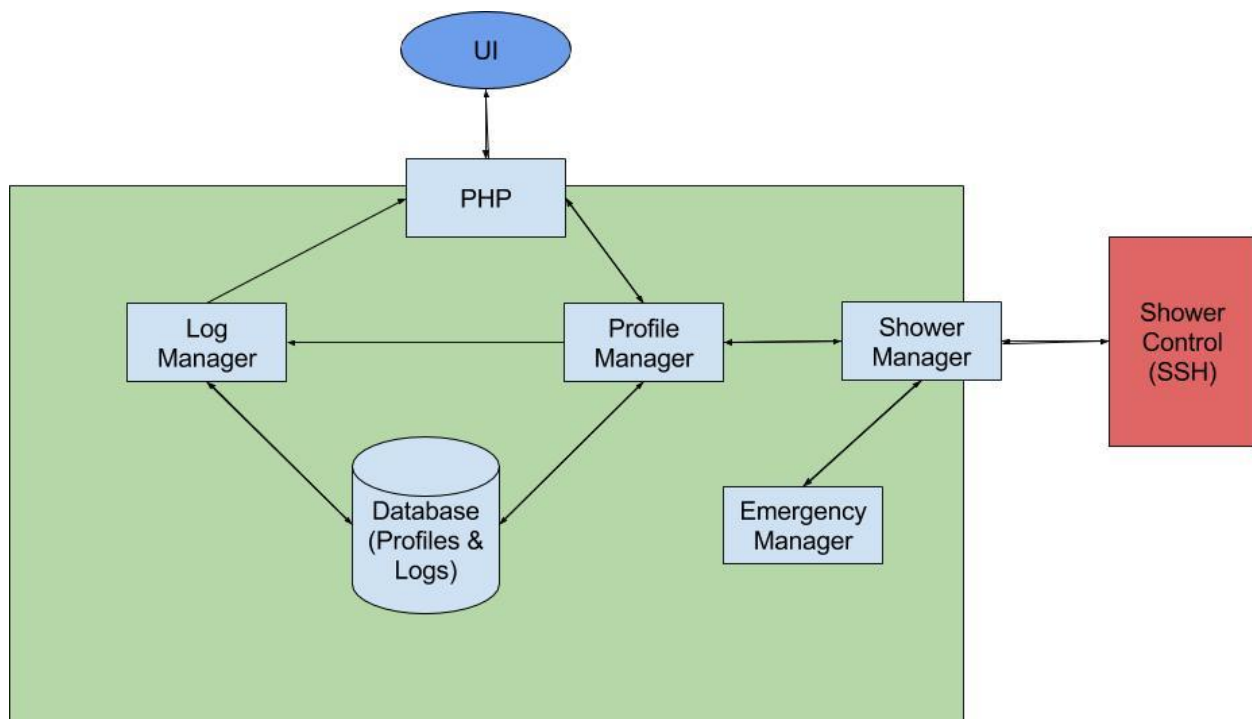
The database will hold all data regarding the logs, profiles, and main system account. The database entries for profiles will hold the most interesting data as it will relate to shower positions, temperatures, and other profile related components.

The shower manager will handle all passage of communication between the WCS and the shower system. It will largely deal with managing system calls, structuring of data, and validation of data.

The emergency manager will be in control of the communication of the emergency safety net that will exist for the shower system. It will handle the sending of text messages, phone calls, or other forms of communication to alert a safety contact of a situation.

## Component Interactions

The diagram below details the components of the WCS and their main interactions with one another. The direction of the arrows indicate which component talks to which. Note that all components touching or within the green square are considered to be managed using PHP, while the database will be a MySQL instance. The UI will consist of multiple technologies, and the shower control is a C program called by all things that use the shower system, and sends its commands to the shower using SSH system calls.



At the top level, the input of the user in the UI is packaged and sent via PHP to the profile manager, which then passes the information to whichever component is needed. If a log is being requested, the profile manager will alert the log manager and the log manager will send back the requested data (after grabbing it from the database) to the UI.

If the request involves the altering of a profile in any way, then the profile manager itself will manage this and access or alter the database in any way requested by the user. If the shower itself is being controlled through the UI, then the commands will be sent through the profile manager and into the shower manager, which will package the information as it sees fit and send it off out of the WCS and to the shower control C program.

Lastly, if an alert is sent to the shower manager that a user fallen, the emergency manager will be sent this data and it will send messages to its various emergency contacts. It gets its data by being passed the contact info from the shower manager, which in turn gets it through accessing the profile manager.

It is important to note the differences in arrows. While most components have access in both directions, the log manager is an interesting exception. Because it's span of control is so narrow, it does not need to speak with anything but the UI and the database; it only stores logs or displays them. However, it must be able to be called by the profile manager so as to receive requests from the user.

## **Sending a Shower Command**

Because sending a command to the shower itself is the core functionality of the WCS, it is important to delve deeper into how this happens.

The input of values from the user can come in many forms. They can either come as an entire set (head position, temperature, and mode), an incomplete set (head position and mode) or a single command (move x head position +5). If a shower is being started for the first time from a profile, then the entire set of commands will be sent through the system. However, if only certain aspects are being changed while the shower is running, then only those values will be sent through.

Once the values have been input, client-side PHP form validation will be run to make sure the commands being sent in meet the basic requirements of the shower system. If they have been accepted, the values will be sent to the profile manager, which will send the information not only to the shower manager, but also to the log manager for logging purposes.

After the shower manager has received the changes, it will process the values and convert them (if necessary) to the correct format needed by the C program. The shower manager will then use PHP system calls (exec) to call the C program with the values passed through. After it has sent these off, it will check for any errors sent back from the C program. If all is well, it will pass back a confirmation through the chain to the UI for the user to see. If there was an error, the shower will completely shut off and a detailed error message will be passed to the UI.

## **Database Structure**

It is worth detailing, at least briefly, a more in depth review of the database access.

The system design was made with the thought that it is best to keep as few components as possible speaking with the database; that is the reason why the profile manager is one of the most integral parts of the system; it handles every database call or manipulation besides log management.

This design keeps the database structure fairly simple; it has log-related tables and profile-related tables. The profile tables hold all information regarding saved shower settings and profile credentials, while the log tables keep any information that is sent to it from the profile manager regarding usage statistics of the shower.