# FOODPORT DESIGN DOCUMENT

Foodport users are essentially in one of two states: (State 1) Logged Out, and (State 2) Logged In. The implementation for State 1 is relatively state forward; the entire state is composed of a single HTML webpage (all sections of State 1 are on the same webpage).

The more juicy part of the website lies in the implementation of State 2. Let's start with the background SQL databases. The foodport website is linked to one database (named fp for final project) which is composed of three tables: orders, port_history, and users. The orders table stores orders, the port_history table store port history, and the users table stores all users and passwords. The general login-logout implementation (with login.php, logout.php) of the webpage was taken directly from pset7 (the CS50 Finance pset); information is passed from the forms to the server by POST.

Registration is relatively straightforward. The user enters a name, email address, username, password and password confirmation to the index.html webpage. The email addresses and usernames are unique in the SQL database; there cannot be more than one user with the same email or username. Everything is checked for validity, and if any errors occur, an error message is shown. The program then adds the hashed password values to the database, and logs the user in (setting $_SESSION["id"] and $_SESSION["username"]).

Ordering groceries is very similar, in that the HTML form (order_form.php) passes information to order.php, and after athentication, the order is added to the database. When entering the user's address, I used google maps to create a drop down menu (as in pset8).

The order database stores various fields:

1. primary id, that auto-increments for every order
2. user_id, which stores the id of the user submitting the order
3. shoppingList, which stores the users shopping list as entered
4. time, which stores the order time in a timestamp format
5. address, which stores the desired delivery address
6. longitude and latitude, the location of the delivery address
7. fulfilled, which is 0 if the order hasn't been fulfilled, else the id of the deliverer who has fulfilled the order if the order has been fulfilled
8. time_fulfilled, which stores the time at which the order was fulfilled

Note that longitude and latitude weren't asked of the user; I used a google maps library that provides the longitude and latitude given a location. Following the order, order_view.php is rendered which displays the order confirmation.

The implementation for delivery/porting is a bit more complex. The user enters into port_form.php his/her home location, the number of orders he/she is willing to fulfill (1-5), and a porting radius (10-50 miles). From there, the information is passed to port.php, which queries all unfulfilled orders somewhat close (within 1 degree longitude and latitude) to the home location. Then using a point to point distance formula, the program finds all unfulfilled orders within the porting radius and chooses orders up to the users willingness level specified. We then provide a PortPass (in

port_view.php), which is essentially a deliverer's confirmation that specifies the orders that the deliver must fulfill. The PortPass contains information that will help the porter attend to the order.

The history implementation involves two parts: the order history and the porter history. The order history is generated by querying all rows from the order database that have matching id's (in history.php which renders history.php). The porter history is generated by first querying the user's port history from the port_history SQL table, and then using that information and another query to orders (searching by the fulfilled keyword) to display the information.

Finally, the homepage shows a google map centered around the user's location (similar to the one in pset8) that queries all fulfilled orders in the given day and displays them as markers on the map. An event listener is added to each marker and, using an associative array (LatLng->String), we can display the shopping lists that were fulfilled all around the world.

Logout is again is similar to the pset7 implementation of logout (see logout.php).