Buzzword Progress Report

Stage Two

## Section 1:

By this point in the original plan we had aimed to implement all of our ‘must have’ functional requirements. At the time of writing this has not been achieved with only F-UR-1.2 and 1.3 being fully implemented. However, we are working hard to finish implementing the functional requirements and expect to be finished with these in two weeks or so.   
  
At this point, we have a user interface for our customer component and almost all functionality implemented with only some bugs related to requesting the data from the server causing an issue.

A waiting component, which displays the menu to the waiters from a JSON file, and allows waiters to select items from a menu and place it in a cart. We still have to implement the ability to amend existing orders, as well as add modifications to menu items, as well as placing an order before this component will be complete.   
  
A kitchen component, which displays items in a respectable format to the kitchen staff. We still need to implement receiving orders from the server, and adding preparation estimates before this component will be complete.   
  
During stage three, we will work to first get the functional requirements for these components implemented and to have the components communicating with the server correctly, then begin improving the UI for these components.

## Section 2:

The system is designed as three separate components; the kitchen, waiting, and customer components; which all communicate by passing a JSON file to each other via the server.  
  
The customer interface design is based around two text boxes where the user will input their keyword, and table number. These two details are combined and sent to the server when the user submits the form to look up their order. This is then passed back to the component as a JSON file and used to generate the orders related to the keyword + table number combination the user provided. For example: John Smith at Table 13 would receive all orders titled Smith13 in the database.

The waiting interface design is based around table views listing the items on the menu, containing information about each item on the menu, including description and cost, and a button which allows the waiter to add the item to the cart for the order. In the future, this will also be extended to include the ability to add ‘special notes’ along with the item which will inform the kitchen of any amendments the customer wishes to make to the standard menu item.

The kitchen interface design is based around tables, showing data about orders which have been placed. In the future, this will be stripped down to show only the bare information the kitchen ‘need to know’ in order to improve usability.

Class diagrams, and data flow diagrams related to the system design can be found in the report for Stage 1.

## Section 3:

The primary method of testing the components has been to look at the output provided by the system. As everything is based around generating user interfaces from JSON files, it is intuitive to assess if the system is performing as expected by comparing the output from each component to the JSON file it has received. This has been combined with HTML, CSS, and Javascript syntax checkers, as well as JSON validators to ensure there are no errors in these files.

The server has been tested by generating ‘stub’ test queries in order to ensure that data is passed to and received from the API as expected.