

# **The Beverage Dispenser (TBD)**

## **Assurance Test Plan**

**18-549: Embedded Systems Design, Spring 2013**

**Lance Christian, Mike Gardener, Arjuna Hayes, Lawrence Tsang**

## System Testing and Experimentation

### 1) Make a single ingredient drink (*Fault Free Test Case*)

*Why* This tests for the baseline use of our product

*How* From web-app send a request for a preset drink that requires only one fluid ingredient.

### 2) Make a single ingredient drink with no cup and handle exception properly (*Failure-Injected Case*)

*Why* This tests for the presence of a highly possible human error when using the product

*How* From web-app send a request for a preset drink that requires only one fluid ingredient but do not have cup in place.

### 3) Make a multiple ingredient drink with ice (*Fault Free Test Case*)

*Why* This tests for the more complex feature of our product involving the ice factor

*How* From web-app send a request for a preset drink that requires ice and at least one other ingredient

### 4) Make a hot water based mixture drink (*Fault Free Test Case*)

*Why* This tests for the more complex feature of our product involving heated water and powder dispensing as well as the blender capabilities.

*How* From web-app send a request for a preset drink that requires at least hot water and a solid powder.

### 5) Make a multiple ingredient drink and turn off the scale during pouring – handle exception properly (*Failure-Injected Case*)

*Why* This tests for the possible error of the scale turning off or failing

*How* From web-app send a request for a preset drink that requires multiple ingredients. After the first ingredient is dispensed, turn off the scale while it is dispensing the second ingredient.

## **Test Metrics**

### *Response Time*

Latency from time drink order is sent to time TBD begins making the order

### *Dispense Time*

Latency from drink order is started by TBD to time drink is ready

### *Drink Quality*

10 point scale of quality of drink made by machine as compared to quality of an identical handmade version of the drink. Error in the distribution of certain quantities will also be a comparison such as if TBD was supposed to dispense 2 ounces and it actually dispensed 3.

## **Importance of Metrics**

### *Response Time*

If our web application is extremely slow, we may need to use a method of interfacing at the TBD rather than ordering via a web browser. The product will be useless if it takes too long to respond to a request.

### *Dispense Time*

If TBD takes more than a minute to dispense any given drink, we need to streamline the process to make it more desirable. Customers will not want the product if they have to wait 5 minutes for a beverage they could make in 2 minutes themselves.

### *Drink Quality*

A responsive TBD that produces undesirable drinks is also a useless product. Comparing the quality of the drinks to handmade drinks and measuring the dispensing error can help us determine if it is producing high quality beverages.

## **Reproducibility of Tests**

Reproducibility of tests will be achieved by recording information related to hardware configuration and software version used as well as the ingredients present in TBD.

## **Automation of Tests**

Our tests cannot be automated as each drink requires the user remove the filled cup and place a new cup in its place. Consequently, our tests must be thorough and thoughtful to ensure we make the most use of our time.

# **Experimental Design**

## **Hypothesis**

To prove we have created a fast-responding, reliable, and high quality drink and solid dispensing product.

## **Metrics**

Response Time, Dispense Time, and Drink Quality

## **Workload**

Be able to consistently deliver high quality drinks at the rate of roughly 1 drink per minute for at least an hour. We are not including time spent waiting for user to remove or add a cup or send a drink request as part of this requirement.

## **Parameters**

Number of ingredients, web browser used to send request, and number of requests sent.

## **Test Run**

Repeatedly send requests to TBD from various web browsers by varying the number and type of ingredients as well as how many requests are being sent.

# **Experimental Results**

We intend to use three graphs to present our data. Response time and dispense time will be on box and whisker plots with the time on the y-axis and number of requests being on the x-axis for response time and number of ingredients being on the x-axis for dispense time. Drink quality will also be represented on a basic plot with the y-axis being the average quality difference between TBD made drinks and handmade drink and the x-axis being the ingredients used for the test.