

SENIOR PROJECT

REQUIREMENTS DOCUMENT

---

# DRINC

## Dynamically Refreshing Interplexing Number of Cordials

---

Brandon Arnold  
BArnold@vtc.edu  
Owen Ledvina  
OLedvina@vtc.edu

Hoang Phan  
HPhan@vtc.edu  
Kyle Timins  
KTimins@vtc.edu

January 28, 2014

## Revision History

Revision Number	Date	Comment
1.0	Oct. 04, 2013	First version of document available

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Purpose . . . . .	1
1.2	Definitions . . . . .	1
1.3	Objective . . . . .	1
1.4	Potential Customers of the Product . . . . .	1
1.5	Assumptions and other Relevant Facts . . . . .	1
<b>2</b>	<b>Functional Requirements</b>	<b>2</b>
2.1	Hardware Considerations . . . . .	2
2.1.1	Frame . . . . .	2
2.1.2	Power Supply . . . . .	2
2.1.3	Back End Control Systems . . . . .	2
2.1.4	Drink Transport Track . . . . .	2
2.1.5	Track Servos . . . . .	2
2.1.6	Flow Meters . . . . .	3
2.1.7	Valves . . . . .	3
2.2	Software Considerations . . . . .	3
2.2.1	Website . . . . .	3
2.2.2	Android App . . . . .	5
2.2.3	Backend Server . . . . .	7
<b>3</b>	<b>Nonfunctional Requirements</b>	<b>8</b>
3.1	Hardware Considerations . . . . .	8
3.1.1	Valves . . . . .	8
3.1.2	Frame . . . . .	8
3.2	User Interface . . . . .	8
3.2.1	Website . . . . .	8
3.2.2	Android App . . . . .	9
3.2.3	Backend Server . . . . .	9

<b>4</b>	<b>Use Cases</b>	<b>10</b>
4.1	Logging In . . . . .	10
4.1.1	Actors . . . . .	10
4.1.2	Assumptions . . . . .	10
4.1.3	Procedure . . . . .	10
4.1.4	Expected Outcome . . . . .	10
4.2	Logging Out . . . . .	11
4.2.1	Actors . . . . .	11
4.2.2	Assumptions . . . . .	11
4.2.3	Procedure . . . . .	11
4.2.4	Expected Outcome . . . . .	11
4.3	Create a Custom Drink . . . . .	11
4.3.1	Actors . . . . .	11
4.3.2	Assumptions . . . . .	11
4.3.3	Procedure . . . . .	12
4.3.4	Expected Outcome . . . . .	12
4.4	Select a Drink . . . . .	12
4.4.1	Actors . . . . .	12
4.4.2	Assumptions . . . . .	12
4.4.3	Procedure . . . . .	12
4.4.4	Expected Outcome . . . . .	13
4.5	Delete a Drink . . . . .	13
4.5.1	Actors . . . . .	13
4.5.2	Assumptions . . . . .	13
4.5.3	Procedure . . . . .	13
4.5.4	Expected Outcome . . . . .	13

# 1 | Introduction

## 1.1 Purpose

This document describes the requirements of the DRINC Project.

## 1.2 Definitions

**DRINC** Dynamically Refreshing Interplexing Number of Cordials

## 1.3 Objective

The DRINC Project objective is to automate and simplify the process of creating a mixed “Drinc”. The goal for DRINC is to serve drinks while automating backend information such as volume used, and others.

## 1.4 Potential Customers of the Product

People who are unable to create their own drinks due to physical disabilities, or other reasons. High-end bars and restaurants who would like reduce the workload involved in making drinks.

## 1.5 Assumptions and other Relevant Facts

1. Customers meet or are over the legal age.
2. Operators will legally be able to serve any alcohol mixed into drinks.

## 2 | Functional Requirements

### 2.1 Hardware Considerations

#### 2.1.1 Frame

The frame must securely hold 9 liters of mixers, glass enclosures, as well as the back end control systems and drink transport track.

#### 2.1.2 Power Supply

A power supply capable of powering the back end control system, drink transport track, valves, and flow meters must be included.

#### 2.1.3 Back End Control Systems

A microcontroller with enough I/O pins to monitor a flow meter and valve for every mixer, as well as the servos controlling the drink transport track and a data connection to the front end system must be used.

#### 2.1.4 Drink Transport Track

The track must be able to safely, securely, and accurately move a cup to any point on a square grid.

#### 2.1.5 Track Servos

The transport track requires at least two servos, which must be strong enough to move a full pint of mixed drink reliably.

### 2.1.6 Flow Meters

Flow meters sensitive enough to measure a minimum of 50ml over a very short time span (around a second) must be installed on each bottle.

### 2.1.7 Valves

Each mixer must have a valve that can be quickly turned on and off via the back end control system.

## 2.2 Software Considerations

### 2.2.1 Website

- 2.2.1.1. The ability to log in based off of authentication held in the backend database.
- 2.2.1.2. The ability to log off of the account.
- 2.2.1.3. Once logged in, the user will be presented with the main menu, which presents the user with the options “Create a Custom Drink”, “Select a Drink”, and “Most Drank”.

#### Create a Custom Drink

- 2.2.1.1. When “Create a Custom Drink” is selected, the user shall be presented with the options “CANCEL” and “SAVE”, a text field to name the drink, as well as an option to add an ingredient to the recipe. This includes an ingredient list and an amount list.
- 2.2.1.2. The ingredient list shall hold all available ingredients.
- 2.2.1.3. The amount list shall hold a list of integers, ranging from 1 to 10.
- 2.2.1.4. Additional ingredients can be added by clicking the “Add Ingredient” button which creates another row allowing the user to select another ingredient, and it’s amount, to the list. If, at any time, the amounts of all the ingredients combined is greater than (but not equal to) 10, an error message shall appear stating that there are too many ingredients and the last ingredient amount modified will be reverted back to it’s original state.
- 2.2.1.4.1) The “CANCEL” button shall return the user to the main menu.

- 2.2.1.4.2) The “SAVE” button saves the drink name with it’s ingredients and their amounts to the database.

### **Select a Drink**

- 2.2.1.1. When “Select a Drink” is selected, the user shall be presented with a list of all drinks created under that user, as well as a list of predefined drinks. There shall also be two buttons: “Create Drink” and “Delete Drink”.
- 2.2.1.2. The first drink in the list will be marked as “Selected”.
- 2.2.1.3. When the user clicks/selects a drink, it will be marked as “Selected”.
- 2.2.1.4. The “Create Drink” button shall popup a window with a confirmation message and the buttons “Create” and “Cancel”.
  - 2.2.1.4.1) If the user clicks “Cancel”, the window will close.
  - 2.2.1.4.2) If the user clicks “Create”, the window will close and another window will appear stating the drink is being made. The current drink that is marked as “Selected” shall then be created by DRINC. Once the drink is created, the current window will close and a message stating that the drink has been completed will appear, then close.
- 2.2.1.5. The “Delete Drink” button shall popup a window with a confirmation message and the buttons “Delete” and “Cancel”.
  - 2.2.1.5.1) If the user clicks “Cancel”, the window will close.
  - 2.2.1.5.2) If the user clicks “Delete”, the current drink that is marked as “Selected” shall then be deleted from the server. Once the drink is deleted, the current window will close and a message stating that the drink has been deleted will appear, then close.

### **Most Drank**

- 2.2.1.1. When “Most Drank” is selected, the user shall be presented with a list of drinks, similar to when “Select a Drink” is selected, that shall be sorted based on how often the user has created each drink, from greatest to least. If a drink has not been created (created 0 times), it shall not appear on this list. There shall also be a button labeled “Create Drink”.
- 2.2.1.2. The first drink in the list will be marked as “Selected”.
- 2.2.1.3. When the user clicks/selects a drink, it will be marked as “Selected”.



2.2.1.4. The “Create Drink” button shall popup a window with a confirmation message and the buttons “Create” and “Cancel”.

2.2.1.4.1) If the user clicks “Cancel”, the window will close.

2.2.1.4.2) If the user clicks “Create”, the window will close and another window will appear stating the drink is being made. The current drink that is marked as “Selected” shall then be created by DRINC. Once the drink is created, the current window will close and a message stating that the drink has been completed will appear, then close.

### **2.2.2 Android App**

2.2.2.1. The ability to log in based off of authentication held in the backend database.

2.2.2.2. The ability to log off of the account.

2.2.2.3. Once logged in, the user will be presented with the main menu, which presents the user with the options “Create a Custom Drink”, “Select a Drink”, and “Most Drank”.

#### **Create a Custom Drink**

2.2.2.1. When “Create a Custom Drink” is selected, the user shall be presented with the options “CANCEL” and “SAVE”, a text field to name the drink, as well as an option to add an ingredient to the recipe. This includes an ingredient list and an amount list.

2.2.2.2. The ingredient list shall hold all available ingredients.

2.2.2.3. The amount list shall hold a list of integers, ranging from 1 to 10.

2.2.2.4. Additional ingredients can be added by clicking the “Add Ingredient” button which creates another row allowing the user to select another ingredient, and it’s amount, to the list. If, at any time, the amounts of all the ingredients combined is greater than (but not equal to) 10, an error message shall appear stating that there are too many ingredients and the last ingredient amount modified will be reverted back to it’s original state.

2.2.2.4.1) The “CANCEL” button shall return the user to the main menu.

2.2.2.4.2) The “SAVE” button saves the drink name with it’s ingredients and their amounts to the database.

**Select a Drink**

- 2.2.2.1. When “Select a Drink” is selected, the user shall be presented with a list of all drinks created under that user, as well as a list of predefined drinks. There shall also be two buttons: “Create Drink” and “Delete Drink”.
- 2.2.2.2. The first drink in the list will be marked as “Selected”.
- 2.2.2.3. When the user clicks/selects a drink, it will be marked as “Selected”.
- 2.2.2.4. The “Create Drink” button shall popup a window with a confirmation message and the buttons “Create” and “Cancel”.
  - 2.2.2.4.1) If the user clicks “Cancel”, the window will close.
  - 2.2.2.4.2) If the user clicks “Create”, the window will close and another window will appear stating the drink is being made. The current drink that is marked as “Selected” shall then be created by DRINC. Once the drink is created, the current window will close and a message stating that the drink has been completed will appear, then close.
- 2.2.2.5. The “Delete Drink” button shall popup a window with a confirmation message and the buttons “Delete” and “Cancel”.
  - 2.2.2.5.1) If the user clicks “Cancel”, the window will close.
  - 2.2.2.5.2) If the user clicks “Delete”, the current drink that is marked as “Selected” shall then be deleted from the server. Once the drink is deleted, the current window will close and a message stating that the drink has been deleted will appear, then close.

**Most Drank**

- 2.2.2.1. When “Most Drank” is selected, the user shall be presented with a list of drinks, similar to when “Select a Drink” is selected, that shall be sorted based on how often the user has created each drink, from greatest to least. If a drink has not been created (created 0 times), it shall not appear on this list. There shall also be a button labeled “Create Drink”.
- 2.2.2.2. The first drink in the list will be marked as “Selected”.
- 2.2.2.3. When the user clicks/selects a drink, it will be marked as “Selected”.
- 2.2.2.4. The “Create Drink” button shall popup a window with a confirmation message and the buttons “Create” and “Cancel”.
  - 2.2.2.4.1) If the user clicks “Cancel”, the window will close.

- 2.2.2.4.2) If the user clicks “Create”, the window will close and another window will appear stating the drink is being made. The current drink that is marked as “Selected” shall then be created by DRINC. Once the drink is created, the current window will close and a message stating that the drink has been completed will appear, then close.

### **2.2.3 Backend Server**

- 2.2.3.1. Hold drink information and send to DRINC
- 2.2.3.2. Ability to SSH into machine for maintenance or configuration
- 2.2.3.3. The server information must be backed up every Wednesday, at 9:00AM.
- 2.2.3.4. Keep track of the drinks consumed by the user during the current period of time.

## 3 | Nonfunctional Requirements

### 3.1 Hardware Considerations

Most hardware should emphasize low cost.

#### 3.1.1 Valves

Valves should be made of plastic or a non copper or copper alloy metal so as not to kill the user via copper poisoning.

#### 3.1.2 Frame

Frame should be modular and easily disassemble for transportation.

### 3.2 User Interface

#### 3.2.1 Website

- 3.2.1.1. Shall allow the user to do any task in the least amount of click/push/touch
- 3.2.1.2. The site should look visually appealing, with lack of “clutter”
- 3.2.1.3. The website will be written in html and php
- 3.2.1.4. The website database will be managed with postgresQL
- 3.2.1.5. The ingredient list and amount list will be created using HTML <select> tag
- 3.2.1.6. When a drink is labeled as “Selected”, it will be highlighted using Javascript

**3.2.2 Android App**

3.2.2.1. The Android app should look very similar to the website.

3.2.2.2. Allow the user to log in via a wireless technology

**3.2.3 Backend Server**

3.2.3.1. Any updates must be started on Wednesday after 10:00AM and finish before 12:00PM.

3.2.3.2. The server must handle all login and logout requests in under 200ms.

3.2.3.3. Ability to add a drink recipe with ease

## 4 | Use Cases

### 4.1 Logging In

#### 4.1.1 Actors

4.1.1.1. User

4.1.1.2. System

#### 4.1.2 Assumptions

4.1.2.1. User already has valid login information.

4.1.2.2. User is at the login page.

#### 4.1.3 Procedure

4.1.3.1. System prompts user for login information

4.1.3.2. User enters correct login information

4.1.3.3. User clicks “Login”

4.1.3.4. The system logs the user in

#### 4.1.4 Expected Outcome

4.1.4.1. The user is now logged in.

## **4.2 Logging Out**

### **4.2.1 Actors**

4.2.1.1. User

4.2.1.2. System

### **4.2.2 Assumptions**

4.2.2.1. User is already logged in

### **4.2.3 Procedure**

4.2.3.1. User clicks the “Logout” button

4.2.3.2. The system logs the user out

4.2.3.3. The system prompts the user for login information

### **4.2.4 Expected Outcome**

4.2.4.1. The user is now logged out and is prompted to login.

## **4.3 Create a Custom Drink**

### **4.3.1 Actors**

4.3.1.1. User

4.3.1.2. System

### **4.3.2 Assumptions**

4.3.2.1. User is already logged in

4.3.2.2. User is already at the create a drink page

### **4.3.3 Procedure**

- 4.3.3.1. System presents user with a list of available options to make a drink
- 4.3.3.2. User selects which drinks to use
- 4.3.3.3. User inputs how much of each drink to use
- 4.3.3.4. User chooses a name for the new drink
- 4.3.3.5. User clicks “Create Drink”
- 4.3.3.6. The system stores the new drink in the database.

### **4.3.4 Expected Outcome**

- 4.3.4.1. The drink is correctly placed into the database for future retrieval

## **4.4 Select a Drink**

### **4.4.1 Actors**

- 4.4.1.1. User
- 4.4.1.2. System

### **4.4.2 Assumptions**

- 4.4.2.1. User is logged in
- 4.4.2.2. There are valid drinks in the database

### **4.4.3 Procedure**

- 4.4.3.1. System prompts user for drink selection
- 4.4.3.2. User selects drink
- 4.4.3.3. User clicks “Select”
- 4.4.3.4. The system makes the drink and increments the number of drinks consumed
- 4.4.3.5. User consumes drink



#### **4.4.4 Expected Outcome**

4.4.4.1. The drink is created and consumed. User is positively affected by drink.

### **4.5 Delete a Drink**

#### **4.5.1 Actors**

4.5.1.1. User

4.5.1.2. System

#### **4.5.2 Assumptions**

4.5.2.1. User is logged in

4.5.2.2. There is already at least one drink in the database

#### **4.5.3 Procedure**

4.5.3.1. System displays a list of saved drinks

4.5.3.2. User clicks “delete this drink”

4.5.3.3. The system pops up with a confirmation “Are you sure you want to delete drink X?”

4.5.3.4. The user clicks “yes” or “no”

4.5.3.5. If the user clicks “yes” the system drops the drink from the database

#### **4.5.4 Expected Outcome**

4.5.4.1. The drink is no longer listed under saved drinks