

DRINC

Dynamically Refreshing Interplexing Number of Cordials

Brandon Arnold

Hoang Phan

Owen Ledvina

Kyle Timins

December 6th, 2013

Table of Contents

- 1 Functional Requirements
- 2 Non-Function Requirements
- 3 Interfaces
- 4 Server
- 5 What We've Done
- 6 Conclusion

Introduction

- What is DRINC?
- Interfaces
- Server
- Pouring Device

Table of Contents

- 1 Functional Requirements
- 2 Non-Function Requirements
- 3 Interfaces
- 4 Server
- 5 What We've Done
- 6 Conclusion

Hardware Requirements

- Frame: Must hold:
 - 9 liter sized glass bottles, control system, transport track
- Power Supply:
 - Be able to power all systems in the DRINC
- Back End Control Systems:
 - Be able to control all mixing hardware with one micro processor
 - Be able to run main system on one microprocessor and interact with other microprocessor

Hardware Requirements Cont.

- Drink Transport Track:
 - Transport cups safely, securely, and accurately on a square grid
- Track Servos:
 - Two servos strong enough to move a full pint of liquid and glass reliably
- Valves:
 - Installed on each bottle
 - Ability to turn on and off quickly by backend system to pour 40ml parts

Software Requirements

- Website:
 - Log on and off authentication using backend database
 - After successful authentication, user is presented with a menu:
 - Create a Custom Drink
 - Select a Drink
 - Most Drank
 - Delete a Drink

Software Requirements Cont.

- Backend Server:
 - Hold drink information and send to DRINC
 - Ability to SSH into machine for maintenance or configuration
 - Keep track of the drinks consumed by the user during time period

Table of Contents

- 1 Functional Requirements
- 2 Non-Function Requirements**
- 3 Interfaces
- 4 Server
- 5 What We've Done
- 6 Conclusion

Non-Functional Requirements

- Valves:
 - Should be made of plastic or non-copper alloy metal
- Frame:
 - Should be modular and easily disassemblable
- Server:
 - Should be able to handle all login and logout requests in under 200ms
 - The server information must be backed up every Wednesday, at 0900AM

Table of Contents

- 1 Functional Requirements
- 2 Non-Function Requirements
- 3 Interfaces**
- 4 Server
- 5 What We've Done
- 6 Conclusion

Interfaces

- Web Site:
 - Should all the user to do any task in the least amount of clicks
 - Should look visually appealign, with lack of “clutter”
- Android Device:
 - Follow same UI and visual requirements as the website
 - Will allow the user to log in via a wireless technology

Website

- Django/Python, HTML, CSS, Javascript
- PostgreSQL
- Administrative functions

Android Device

- Nexus 7
 - App will be build in the Android version of Java and XML
 - Device will be attached to the DRINC machine

Android Mock-up

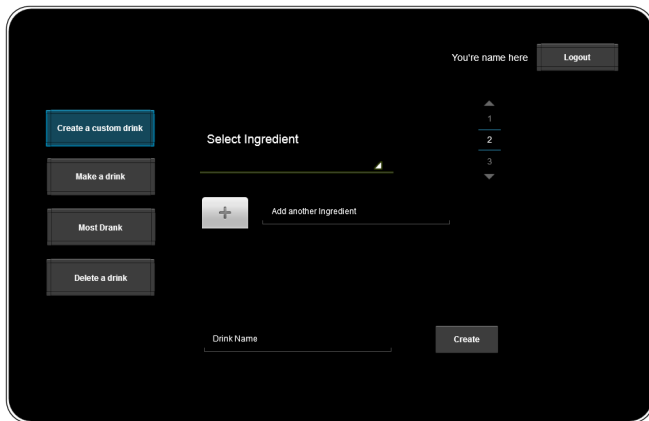


Figure : Example of creating a drink on the Android device

Android Mock-up

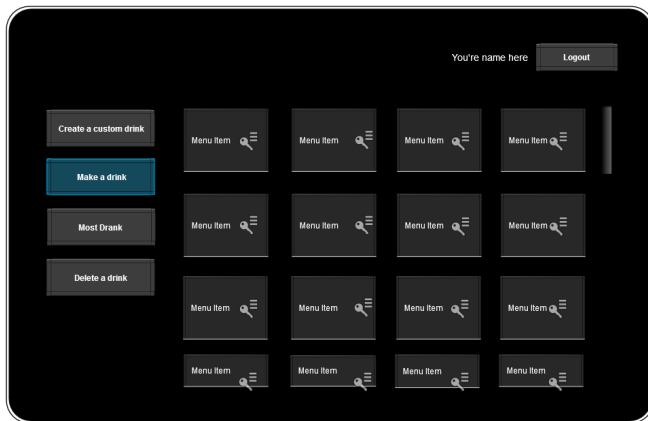


Figure : Example of selecting a drink on the Android device

Table of Contents

- 1 Functional Requirements
- 2 Non-Function Requirements
- 3 Interfaces
- 4 Server**
- 5 What We've Done
- 6 Conclusion

Server

The server will be a Raspberry Pi, with the following specs:

Processor: Broadcom 700MHz

RAM: 512MB

Graphics: VideoCore IV

Hard Drive: 8GB SD Card

OS: Debian Linux (Raspian)

Raspberry Pi Model B

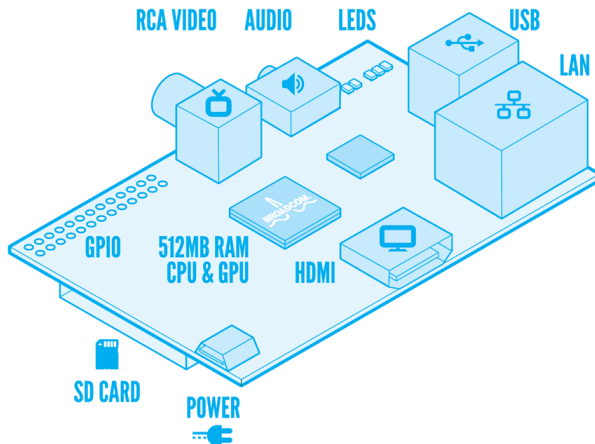


Figure : Diagram of the Raspberry Pi Model B

Server Software

- Raspian
- Apache2
- PostgreSQL

Chasis Mock-up

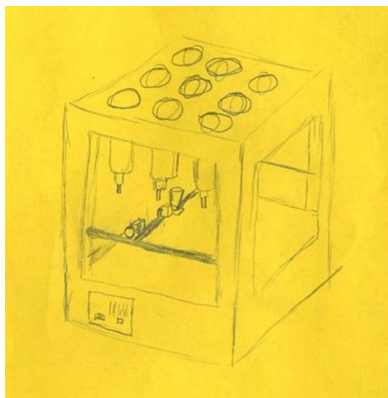


Figure : Diagram of the Chasis

Table of Contents

- 1 Functional Requirements
- 2 Non-Function Requirements
- 3 Interfaces
- 4 Server
- 5 What We've Done**
- 6 Conclusion

What We've Done

- All teammates worked together on all documentation
- Hoang primarily worked on diagrams.
- Raspian installed
- Apache2, PostgreSQL installed

Table of Contents

- 1 Functional Requirements
- 2 Non-Function Requirements
- 3 Interfaces
- 4 Server
- 5 What We've Done
- 6 Conclusion**

Conclusion

Why DRINC?