

UNIVERSITY CALIFORNIA RIVERSIDE

CS179J SENIOR DESIGN PROJECT

SPRING 2017

Smart Fridge

Author:
Stanley COHEN

Professor:
Frank VAHID

June 12, 2017



Abstract

The Smart Fridge is a fridge designed to help users keep track of their fridge contents and buying habits. It consists of three main systems; a system at the fridge to monitor insertions and removals, a system in the cloud to store all data and a smartphone system to communicate with the user.

1 User Guide

Once the fridge is installed, simply plug the Raspberry Pi on to boot up the system. A green LED will light when the database has been properly fetched and the system is ready for inputs. At this point, simply scan your items with the barcode scanner before you put them into or remove them from the fridge. Your insertions/removals will be automatically communicated to the database that is always waiting for more information.

All of your fridge information can be seen in the Android phone app. Once the app has been installed, simply navigate between the *Dashboard*, *Fridge* and *History* tabs to get more information on the contents of your fridge and your past purchases.

2 Technologies and Components

- Raspberry Pi
- Arduino Uno
- IR LED/Receiver
- Firebase NoSQL Database
- Firebase Backend Javascript Cloud Functions
- Android App (built with Java/XML)

3 Demo Video

Demo video can be found [here](#).

4 Source Code

4.1 Raspberry Pi

4.1.1 main.py

This is the main script for the fridge. It is in the *bashrc* file so that when the Pi is booted into terminal mode, it will be run automatically (no monitor needed!). The script fetches the current information from the database and listens for barcode scans from the scanner and IR beam breaks from the Arduino.

If it senses a beam break and then a scan, it will write to the database that the food has been taken out of the fridge. If it senses a scan and then a beam break, it will write to the database that the food has been put in.

4.2 Arduino

4.2.1 IR_sensor.ino

Listens for an IR beam break. On beam break, sends the signal to the Raspberry Pi.

Also looks for a button press. On button press, requests fridge data from Pi to send to Dylan via Software Serial.

4.3 Firebase

4.3.1 index.js

Backend JavaScript used to make reads and writes to the database a LOT easier. Also every minute checks to see if there is expired food.

4.4 Android

4.4.1 MainActivity.java

Main Java file. Attach listeners to the database nodes and handle tab switching along with most everything else in the app.

4.4.2 Food.java

Simple Java class to store a food. Mostly just a bunch of datafields. The only complicated thing here is the string to date converter that matches the default Friebase format.

4.4.3 FoodListAdapter.java

Custom ListView adapter for displaying lists of food. Takes an array of arrays of foods and ensures they're displayed properly when their views are called. Also added a function, *getUsage* for getting the number of purchases on any given day of the week to make it easy to create the graph.