Concept of Operations
Help's Kitchen

COP4331 : Spring 2017

Group 16

- Nathaniel Bates
- Carlos Castro
- Stephen Davis
- Brandon McNamara
- Lucas Plager
- Stephen Ulmer

Contents of this Document

The Current System

In this day and age, most restaurant management software is a medley of unrelated programs and methods that are all used to create a single ecosystem. Management uses one program to monitor staffing roles and responsibilities, another to set prices and menu items, and others for customer relations. Kitchen staff has their own software to determine what meals to prepare and when, while wait and hosting staff have their own unique software or none at all. Support and licensing costs build with each new program added, and in order to manage a restaurant efficiently, many tasks and activities have to be reviewed and accessed separately.

The Proposed System: Motivation

Our system is a single web application to manage and maintain all parts and aspects of the restaurant ecosystem. It centralizes all activities and procedures with seamless compatibility and less hassle. Jailed accounts ensure each employee has access to their specific duties and responsibilities free of interference from other tasks. A single suite permits cheaper deployment and the web platform ensures it will run on pre existing hardware as the only requirement is having the web app. Additionally, owners are able to save on costs by allowing employees to use personal devices as most people already own a compatible device, by not purchasing and maintaining system specific hardware.

The Proposed System: Users and Modes of Operation

Help's Kitchen allows four different types of users:

Host

Host accounts have access to the status of each table. Statuses include **full, available,** and **reserved**. Full tables are currently in use by customers, while available tables have no active customers and can be used to seat those waiting. Reserved tables are marked as taken and are unable to seat customers, but do not require attention from other staff members. Table status can only be changed and modified by Host and Manager accounts.

Server

Server accounts have access to the progress of the meal orders from each table. They are able to take orders from the customers and allows them to identify whether or not the customer has received their meal. In order to place the orders, they have access to the full menu and prices in a quick and intuitive format. As they take the orders, the meals are placed into a queue and assigned to the table. They will receive a notification when the meals for one of their tables is marked as Complete by the kitchen staff. Customer orders are available to be modified by Server and Manager accounts.

Kitchen

Kitchen staff accounts have access to view the orders that have been placed by the Server account. Orders placed by the Server are set to **In Progress** by default and can be updated to **Complete** when the meal has been prepared. A timer will start when the food is ordered to ensure that the meal is prepared and served without unwanted delay. Orders are grouped together by table to allow the kitchen staff to notify the server that the entire table's meal is ready to be delivered. The meal status can only be modified by the Kitchen and Manager accounts.

Manager

Manager accounts are the only accounts able to modify the menu items, prices, and view customer reviews. Manager accounts can also view any real time issues such as long waiting times for customers or issues with food, in order to correct the problem. Additionally, they are able to view the facility metrics such as average table wait time in graphical format. Manager accounts can perform all activities of the other accounts as well as create, delete, and modify accounts.

The Proposed System: Operational Scenarios

Help's Kitchen is designed to allow integration into all parts of the restaurant staff duties. A customer can walk in the door and ask for a seat and the Host account is able to view all available tables. Once the customer has been assigned a table, the Server account holders can take the customer's order and push it to the Kitchen staff. There the meal is prepared by the Kitchen staff and delivered by the Server. Once the customer has finished their meal, they have the option to leave a review only viewable by the Manager and the Server.

In the event a customer would like to change their mind regarding something with their meal, they are able to notify the the Server who can update their meal in real time, so long as the meal has not reached **Complete** status. This allows quick and timely updates to help minimize distress and buyer's remorse after ordering.

Most input by users are based on toggles rather than direct edits, greatly reducing the ability for user error. Table statuses are toggled between the three status types rather than having to type in its status with the only custom input being the number of guests to be seated at the table. The Server also has very little direct input as all menu items would be selected from a list that has been predetermined by the Manager user with only small noting ability for the occasional customer preference. Kitchen staff are unable to have any form of input other than setting a meal's status. This helps keep interaction with the system minimal and quick in the high speed kitchen environment. Manager accounts have the most input and the highest risk of incorrect values, centralizing a majority of possible errors to a single user. All menu updates require an "Are you sure?" confirmation before posting, to help prevent user input error.

The Proposed System: Operational Features

Required:

- Access restricted accounts for each of the four users.
- Manager is able to create and remove users
- Customer Experience reviews
- Updatable Table Status

- Simple Server Menu with products, descriptions, and prices to place orders
- Meal preparation progress status.
- Manager metrics in graphical format

Would Like to Have:

- Notifications for the Server and Kitchen accounts when a change has been applied.
- Push notifications/messages from the Manager accounts to other users.
- Employee “Clock-In/Clock-Out” status

The Proposed System: Analysis

Help’s Kitchen will be developed with a GUI written primarily in Java with the JavaFX GUI library and embedded as an object on a webpage. The backend database and account management will be handled within Firebase. As the page will run as a web based Java applet, any browser that supports Java 8 should be able to run the software, though it will require an active internet connection.

While all developers on staff are fluent with Java, there is a learning period associated with Firebase. We are all relatively new to the platform, though there is ample documentation to help our team implement it as smoothly as possible. The biggest alternative to Firebase is Parse Server, but Facebook and other major tech companies are currently in process of moving from Parse to Firebase. By developing with it from the beginning, we’re able to stay ahead of the curve. It hosts all database information on Google servers and integrates easily with Java while Parse primarily supports objective C and JavaScript.

Since Firebase hosts the database and most of its framework online, it forces our software to become entirely internet dependent. Other backend services allow local management and can be integrated into a local intranet, ideally keeping it independent from any provider faults or errors that may cause connection loss. Without a public internet connection, Help’s Kitchen struggles to keep up to date and no accounts would be accessible. As the application is lightweight and browser-based, while an active connection is available, it is extremely quick and easy to use, but only when the infrastructure supports broadband. Additionally, the web application can also be managed with a personal device over a cellular network, such as a 4g LTE connection, from any mobile device in the event of a local outage.