Group 27

Fudo Hyōka

ECS506U Software Engineering Group Project

Problem/Domain Analysis Report

Contents

1. Introduction

The proposed system, a “restaurant management system”, exists within the domain of the day-to-day operation of a restaurant. Specifically, the problem being solved is taking and fulfilling food orders, as well as the management of related inventory. Both are vital for the success of a restaurant, as satisfied customers will result in good reputation and hence in increased customer traffic and profits.

Traditional ordering systems, involving a staff member manually writing orders on paper, are extremely inefficient and give rise to many issues such as customers receiving incorrect dishes. Additionally, restaurants often struggle with accurately managing their ingredient stocks. Ensuring the availability of sufficient resources to prepare popular dishes is an important issue and has to be addressed to avoid loss of customers.

The proposed system will solve these problems with a web interface, providing separate screens for food orders and managing stock levels. The domain analysis has been conducted in order to gain a good understanding of the environment in which the system will operate in, and make better decisions in other stages of the software engineering process.

2. Glossary

Word	Description
Bill	A printed or virtual statement of the money owed for a service received.
Inventory	A complete list of all ingredients held in the restaurant’s storage at any given time. [1] [2] [3]
Menu	A list of dishes available to customers in a restaurant.
Order	An arrangement to receive a specified set of food/drink items.
Payment	The act of a customer paying the restaurant for their bill, either by cash, card, or cheque.
Reservation	Arrangement made in advance to have a table available at the restaurant, at a specified time and for a specified number of customers.
Stakeholder	Individuals or businesses interested in, or affected by, the success of the system.
Supplier	A person or organization that provides products, such as ingredients, enabling the preparation of dishes.
Table	Each restaurant customer is allocated to a table according to party size.

3. Customers and users

The system's users are employees within the restaurant, who will be interacting with the system on a regular basis. The customers of the system are restaurants that need a reliable tool to handle their many concurrent day-to-day activities. A restaurant adopting the system will result in all its employees becoming users of the software.

Users	**Role**	**Stakeholder priority**
Bookkeeper	Request re-stocking of ingredients from suppliers	Primary
Waiter	Need to know what dishes to serve to a table.	Primary
Cooks	Log any ingredient requirements	Primary
Bartender	Process drink orders	Primary

Customers		**Stakeholder priority**
Restaurants	Purchases system for staff members to use	Secondary

Others:		**Stakeholder priority**
Restaurant customers	Purchase food services from the restaurant	Tertiary
Restaurant supplies	Sell ingredients and other stock to the restaurant	Tertiary
Restaurant other staff	Performs miscellaneous tasks such as cleaning	Tertiary

4. The environment

The current management systems involve a machine/device with embedded software that handles the ordering of food, combined with handwritten notes made by waiter and other staff.

The main disadvantage of this environment is lack of flexibility, as the machines' software is embedded and thus neither easily modifiable nor available on alternative devices. Therefore, staff have to work around the machine's limitations and are dependent on the specific machines, which can be inconvenient.

The new system aims to be used remotely, on any device, thanks to its web-based architecture. This allows the environment to not be restricted to tills. The software will consist of web application hosted on Heroku. This entails the creation of html frontend of the web-app, and the use of javascript for certain aspects of the frontend. In turn, the backend will be based on node.js, and can run on virtually any server provider.

5. Tasks and procedures currently performed

This section outlines the different tasks performed by the employees.

A **waiter's** tasks revolve around serving customers, including taking their orders, serving orders to tables, and fulfilling any further requests. The latter may involve taking additional orders or logging complaints (refunding payments). Currently, waiters handle such tasks in any way they see fit, such as by noting orders and complaints on paper. Shortcuts tend to involve waiters memorizing requests, either to save effort or due to an overestimation of their memorisation capability. Once the bill needs to be calculated, waiters compute the payments either mentally or with a calculator.

A **food manager** handles paper documentation of all needed ingredients, and their quantity.

A **receptionist** is involved in table allocations and reservations. This includes checking availability of restaurant tables - according to party size and availability - before allocating customers to tables.

Many of the above tasks can be automated. Taking orders and associating them to particular tables can be managed by an electronic system, as can checking the availability of tables while automatically taking into considerations party size, reservations, and expected duration of a customer's stay in the restaurant. Furthermore, inventory management can also be automated, with a database keeping track of available stock and updating as dishes are prepared. A simple "restock" functionality could also be implemented, allowing restaurant staff to easily order required ingredients. This could even be done automatically by the system when detecting low reserves of some particular inventory items.

Furthermore, the user interface available to staff members could make use of several visual cues to ease their workload. For example, when delivering orders to a table, the system could show the waiter which dishes are destined to each individual at the table.

6. Competing software

The table below outlines the main competitors to the system being designed.

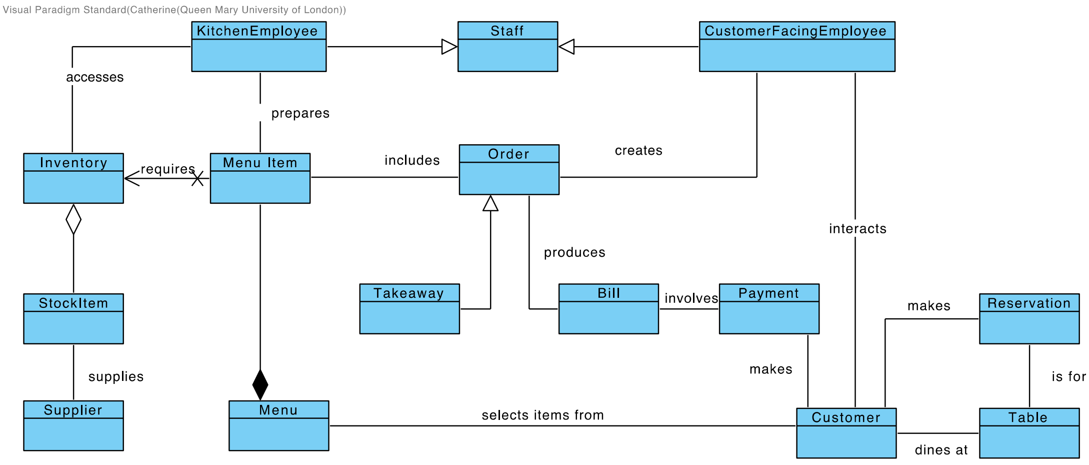
TouchBistro is a software offering commonly used by high-end restaurants. It runs on modern commonly available devices such as iPads, enabling devices to easily communicate with one another. Support is also available at all times, making adoption of the system by a restaurant easy. However, the system has a high price per device, and is limited to specific countries (USA & Canada) [1].

Jolt is a restaurant management application with a focus on customisability, personalised notifications, and easy management of employee tasks. The platform has a £99 monthly subscription fee and is mostly used by cafes and other establishments with relatively small management needs [2].

OpenSimSim is a free, multi-platform and customisable management software that is not specific to restaurant activity. Only one individual may manage the entire system, and needs to make frequent manual updates to task information [3].

7. Domain Model

The domain of the system is graphically presented below.



Bibliography

[1] TouchBistro, “TouchBistro,” 2017. [Online]. Available: https://www.touchbistro.com/. [Accessed 22 01 2017].

[2] Joltup, “Joltup,” 2017. [Online]. Available: https://joltup.com/restaurant-management-software/. [Accessed 20 01 2017].

[3] OpenSimSim, “OpenSimSim,” 2017. [Online]. Available: https://opensimsim.com/. [Accessed 22 01 2017].