Automation of J’s

System Design Documents

Group 3

Ethan Butler

Tuyen Le

Ryan Mobley

Jason Tang

Justin Wade

Table of Contents

[1. Conceptual System Design 1](#_Toc56799592)

[Screen Layouts 4](#_Toc56799601)

[2. Technical Design 6](#_Toc56799605)

[Class Diagram 6](#_Toc56799606)

[Class Specification 7](#_Toc56799607)

[Database Table Descriptions 8](#_Toc56799608)

# 1. Conceptual System Design

Automation of J’s

Staff

Menu

Schedule

Table

Order

Management

Kitchen

Inventory

System

Reports

Finance

Order

Cook

Materials

Goods

Budget

Purchase

### Problem Definition:

The restaurant needs a system to manage all orders, transactions, and management that take place. The goal of this system is to communicate with the kitchen with short but clear interactions, while also saving transactions and interactions for management. It is also to provide a way for management to view the status on staff and other parts of the restaurant.

### System Objectives:

* Makes sure the orders of the restaurant are functional
* Makes sure it is recording transactions
* Track orders to the kitchen once they have been inputted
* Update table status as soon as their status changes
* To be sure that the interactions between waiters, kitchen staff, and hosts/hostesses are short and easily understandable.

### Constraints:

#### External:

* Being sure that inputs can still be done after first input from a customer’s order
* Making sure inventory is in stock for orders

#### Internal:

### Information needed:

* What do orders consist of
* What table does the order belong to
* When a table’s order is finished
* When a table is vacant
* When a table is clean

### Information Sources:

* Manager
* Waiters
* Kitchen/Chefs
* Host/Hostess

### Overall system flow:

When a customer arrives, they are assigned to an available table by a host/hostess, while also assigning a waiter for that table. Once they are seated, the assigned waiter will take the customer’s order, and input it into the menu system through the screen made for waiters. The order will then be displayed on the order queue screen where the kitchen staff will proceed to prepare the meals for the table. Once finished the orders are sent to the tables through their assigned waiters. When a customer has finished their order, it is closed on the system and the system sends the completed order to the completed tab. Then the system sends a signal to send someone to clean that table. Once the table is cleaned the waiters will report it as clean to the system. Therefore, the host/hostess may assign another customer to the table and begin the whole process again.

Management can check the finance of the restaurant and the current budget. If needed management can also purchase goods and inventory to restock the kitchen for future. They can also view reports of staff member as well as add new ones.

### System Inputs:

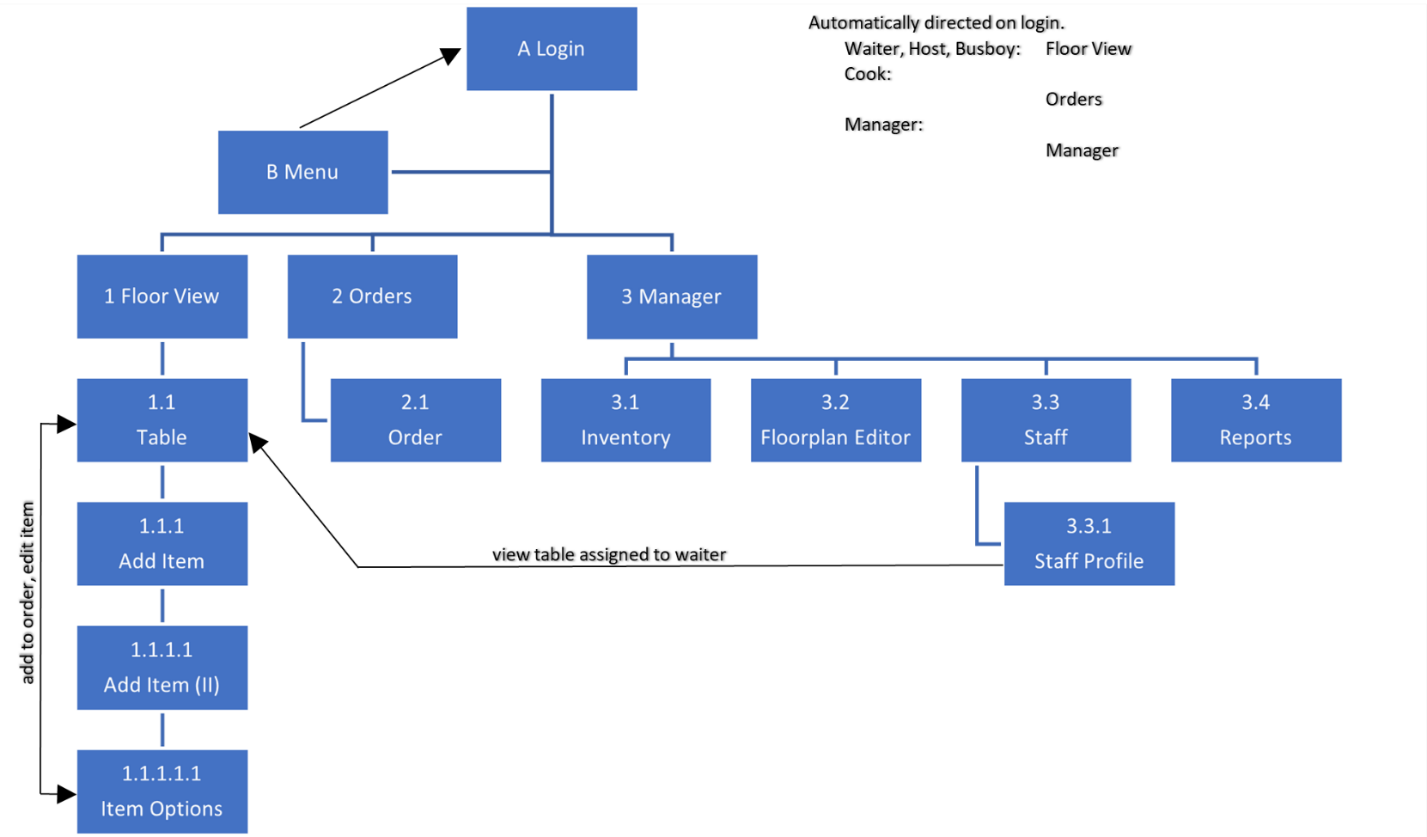
* Order status
* Table status
* Staff status

### System Outputs:

* Order status
* Staff status
* Table status

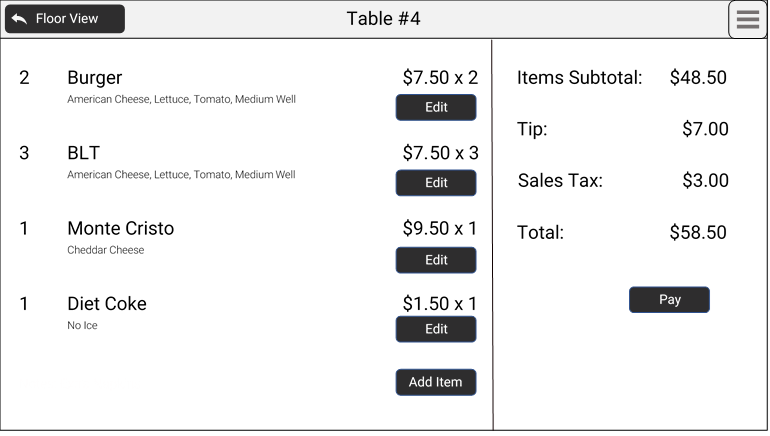
## Screen Layouts

### Navigation Flow



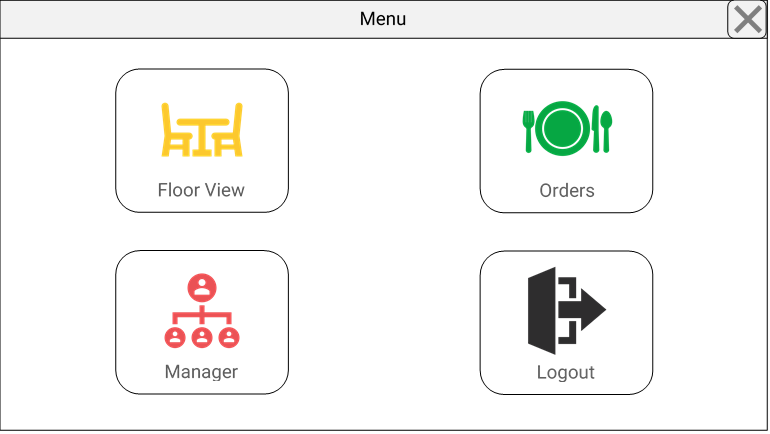
This diagram above illustrates the path through which a user can navigate the software. Screens adjacent to a given screen on this diagram can be accessed from that screen. A user can move to a further screen using the appropriate option within the screen or use the back button to move to the previous screen. Additional navigation options are noted with arrows in the diagram. All screen prototypes can be viewed in the Sprint 1 Prototypes submission.

### Navigation within a Screen



The application has a menu bar spanning the top that displays the name of the screen (above example: details for a selected table). If applicable, a back button is present at the left side of the menu bar, allowing a user to return to the parent screen. A toggleable application menu button is shown on the right side of the menu bar.

### Application Menu



When toggled, the application menu allows a user to navigate to an available default screen (Floor View, Orders, and/or Manager) or to log out.

# 2. Technical Design

## Class Diagram

Diagram

Description automatically generated

## Class Specification

The Application class is a manager for the system. Its methods include generating reports, fetching inventory data from the database, and checking login IDs against the database of employees.

The Employee class is an abstract parent class that holds variables common to all employees, regardless of role.

The Employee class has five subclasses: Busboy, Cook, Host, Manager, and Waiter. Each subclass includes variables or methods that are specific to that type of employee. The Waiter class includes an attribute to track which tables are assigned to that waiter and methods to add/remove assigned tables and retrieve a list of all employees.

The Table class is used to handle data and methods related to a single table in the restaurant. It includes attributes describing its number, size, and position, as well as any orders currently at the table and its state (e.g. occupied, dirty, or ready). It includes methods to change its current state, change the order associated with it, remove an order from the instance, signifying that the customers at that table have left.

The Order class stores data related to orders, and objects of this class are stored within a Table. The Order class includes attributes such as its number, all of the Products that have been ordered, and its status. It includes methods such as adding or removing products to the order.

The Product class has attributes describing the name of the menu item (“product”), options associated with it, and notes. The Product class draws information from the Item class, which relates to the Item table in the database. It has attributes for all the columns of the Item table and has a method to update the database entry as new information is added.

The Report class has attributes for the type of data the report will return and the date for the data. It has methods to generate a report based on these attributes and to format the report for display.

## Database Table Descriptions



Database Table Descriptions

**Order Table**

The order table keeps track of the orders J’s receives from customers. Its attributes are OrderNumber, OrderStatus, OrderDate, OrderTime, ProductName, and TableNumber. The table’s primary key is OrderNumber. The table’s foreign keys are ProductName and TableNumber. These foreign keys provide relationships with the product and table tables.

**Product Table**

The product table keeps track of the products J’s sells on its menu. Its attributes are ProductName, ProductQuantity, and ItemName. The table’s primary key is ProductName. The table’s foreign key is ItemName. This foreign key provides a relationship with the item table.

**Table Table**

The table table keeps track of the tables to dine at inside of J’s. Its attributes are TableNumber, TableStatus, and EmployeeNumber. The table’s primary key is TableNumber. The table’s foreign key is EmployeeNumber.

**Employee Table**

The employee table keeps track of the staff members working at J’s. Its attributes are EmployeeNumber, EmployeeRole, EmployeePhone, EmployeeEmail, EmployeeAddress, EmployeeRate, EmployeeHours, EmployeeAuthorization, and EmployeeNote. The table’s primary key is EmployeeNumber.

**Item Table**

The item table keeps track of the items in J’s inventory. Its attributes are ItemName, ItemCurrentStock, ItemNeededStock, ItemAmountToPurchase, ItemPricePerUnit, and ItemTotalCost. The table’s primary key is ItemName.