Restaurant Automation

Report 1 - Part 2

February 12, 2017

Project Team

Mit Patel

Raj Patel

Nill Patel

Dylan Herman

Prabhjot Singh

Moulindra Muchumari

# All team members contributed equally!

Table of Contents

Customer Statement of Requirements

Running a restaurant with pen and paper in the age of technology is getting very inefficient. In order to keep up with the competition and take advantage of the opportunities provided by new technologies. We would like to have a system that will integrate all the employees and allow for seamless tracking of each task. Automation decreases cost by hiring less people and helps to remove the human error that causes customer dissatisfaction.

Customers don’t have a way to express feedback regarding their experience

Sometimes our customers may have a bad experience at our restaurant that can lead to a loss of customers. If we are aware of such negative experiences, we can prevent them from occurring in the future, thus increasing customer satisfaction. Currently, submitting complaints at our restaurant comes at a cost of time to both the manager and the customer. Usually, our customers do not want to go out of their way to waste time submitting a complaint. Moreover, spending time attending to customer complaints comes at an inconvenience to the manager because he or she must temporarily stop overseeing restaurant operations.

Imagine a full time employee at an office decides to take their one-hour lunch break at our restaurant across the street. The customer arrives at our restaurant, waiting to be seated. A waiter unwelcomingly greets the customer. Usually, such an incident would not be worthy of complaining about to the manager. The customer is seated, and the waiter takes the customer's order. The food arrives, but the customer receives the wrong item. The waiter sighs and brings the food back to the kitchen. Fifteen minutes later, the customer’s food finally comes out. After finishing his/her food the customer pays the bill and leaves the restaurant in a hurry because they had to go back to work. The customer was unable to give the restaurant’s manager feedback about the poor service they received and most likely wouldn’t go back there to eat.

We would prefer a solution that would allow us to easily receive customer feedback without coming at a cost of time to us or the customer. Also, we would like to be able to view and save customer feedback for personal records to prevent any feedback from being missed.

There is a communication issue between the host and the customer to notify the customer when he or she can be seated

When customers arrive at our restaurant, they want to spend less time waiting, and more time dining with their friends and/or family. During busy hours, wait times can be as long as one to two hours long, meaning that the lobby will be crowded. In such circumstances, it is difficult for the host/hostess to notify a customer that their table is ready above all the noise and ruckus. Miscommunication errors can lead to inefficiency in the seating system, thus increasing overall wait time and customer unhappiness.

Assume a family of six people visit our restaurant at 6PM on a Friday night to enjoy a family dinner. The family enters the restaurant and the host informs them that the wait time will be one hour and 30 minutes to be seated. The family decides to take advantage of the long wait time by visiting the mall across the street. One hour later, an unexpected amount of tables were vacant, making room for the family of six. Unfortunately, the family was not present at the time they were at the top of the queue. The host spent 5 minutes looking for the family of six, unable to find them. As a result, the host assigned the table to the next customers in line. The family arrived 10 minutes later, only to learn they had to wait an extra hour.

We would prefer to have a system that will let us contact our customers when there is a wait time. The customers should be able to keep in contact with us to know if they would like to keep their reservation.

Managing customer checks both efficiently and in an organized manner

Using the old fashioned system to generate bills and keep records is cumbersome, inefficient, and unorganized. The current system requires our waiters to manually calculate subtotals for a customer's bill, which can be time consuming and prone to errors. Moreover, keeping records of all checks for future references requires extra work and responsibility for the manager of our restaurant.

Imagine a party of 15 walks into our restaurant, and they order a lot of food. When it comes time for the bill, the waiter has to start manually adding up each order to get the total for the bill. This can take quite some time because the waiter has to make sure the bill is accurate. After the bill is paid, the waiter adds the bill to the daily stack of bills which will then need to be logged.

One possible solution that could help us solve this is a system that will help us remove the manual calculation of bills and store the data on a computer. This will help save time for both the customer and the restaurant.

Inefficiency with table management & cleaning

A common problem we find in our restaurant is the inability to keep track of unoccupied or dirty tables, which leads to inefficiencies in seating our customers. This means we lose potential revenue because our customers experience longer wait times and are not satisfied.

When a customer comes into our restaurant they first approach the host to be seated. The host has to look at the whiteboard and see which tables are available and seat the customer(s) accordingly. At busy times in our restaurant, the customer may have to wait a significant amount of time for a table to become vacant. However, the wait time is longer than it should be because the whiteboard doesn't get updated instantly when the busboy cleans the table.

A solution we believe would solve this problem is to have a system that allows us to view the dining area in realtime. It would display when tables are vacant, occupied or dirty using different colors and patterns to distinguish between the three statuses. This would allow us to view instantaneous updates about the status of the table and decrease wait time and increase customer satisfaction, when compared to the traditional whiteboard method.

Splitting checks & generate bills

When parties finish their meal, there are instances where they forget to mention splitting checks in the beginning and inform the waiter of splitting checks when the total check has already been calculated. In such a case, the waiter would have to ensure the orders of each individual and split the checks accordingly. In such a situation the waiter would have to go back to the cashier and inform them to print individual checks for each customer. Situations like these usually become a hassle for both the waiter as well as the customers. This leads to customer dissatisfaction, adds to the wait time of the customer(s) which in turn makes that table unavailable for a longer period time for the next customer(s).

A possible solution to this should allow us to split the checks whenever necessary, so this way we don't end up wasting time and resources and can seat the incoming party as soon as possible.

Wasted time in relaying the order to the kitchen

In our restaurant, the waiter has to jot down the order on a notepad for the kitchen and take a carbon copy to the cashier. This is very inefficient in the sense that the waiter will have to make multiple trips between the table, kitchen and the cashier. Doing so for multiple tables, takes away time from serving our customer's needs. The kitchen notifies the waiter when the order is ready by ringing a bell, but the waiter is not going to know whether the bell was rung for his order or a different one. This wastes the time of our waiters and causes them to run around for no reason. Also, we had previously lost or damaged the paper copies. Sometimes we have issues reading the waiter's handwriting.

A possible solution is to make this entire process electronic. The waiter could take the order on a tablet or some device instead of jotting down on a notebook. Then, the order could be sent electronically to the kitchen and the cashier. Also, the same system could notify the waiter when his order is ready. We think this would really decrease the amount of trips to be taken by the waiter. Moreover, we think it would be easier for us to process the bills.

Prioritizing and managing the flow of orders in the kitchen

Our kitchen is typically a chaotic place. Most of chaos results from us having to keep track of all the orders and which of our waiters they can from. However, not only do orders have to be served in a first-in first-out fashion, but the time of delivery made needs to dynamically change based on multiple factors such as: fairness (did one table get both their appetizers and entrees, while another table who arrived at the same time got nothing?), is our restaurant having a slow day and they want to keep as many customers in the restaurant as possible to appear "busy", does a customer want their food to be held until they finish their appetizer?, and etc. Thus managing the kitchen is more complex than simply queueing up orders as they come. That's why adding automation can create order in our kitchen.

A solution we believe would solve this problem is to have a system that automatically keeps track of orders as they come in. However, in our restaurant the orders that come first are not always the orders that get dished out first. We would need to have multiple order queues to make sure all tables get the food in a fair manner (appetizers come first, etc...). We would also like the system to automate the delegation of tasks (which cooks cook what). These multiple queues would also be help for customers that push orders back; they should be put behind other orders that have not been dished out yet.

Managing and keeping track of all the raw materials in the kitchen

Our restaurant has problems with tracking raw materials in the kitchen. We usually order the raw materials for the next week so we receive them on Sunday before the week starts. Since we can't know exactly how much we will use, we order more than required and throw away extra materials at the end of the week. We lose a lot of money and materials because of this. We tried to keep track of the raw materials every day by noting them on paper. But, our cooks found it difficult to note down what they used while cooking. Sometimes we were wrong with our estimate and we ran out of materials during the week.

We would like a solution that electronically deducts materials whenever the cooks finish an order. Then, the system could notify us to if any materials are running low, so we could order them on the go.

Planning the menu and activities for the day

Many restaurants plan activities the night before to streamline cooking for the next day. This allows them to serve more customers and reduce the waiting time for customers to receive their food. A lot of restaurants do not plan these activities properly or allocate enough time to finish them the night before. Hence, this reduces productivity next day and increases stress and unhappiness in the kitchen. Moreover, this might directly affect the revenue of the restaurant and decrease customer satisfaction.

A solution is to combine the planning activities with the cooking activities, so that planning for the next day happens concurrently with serving customers. This solution might work because the planning activities can be completed when the restaurant might not be busy. Also, this simplifies managing the kitchen for the Chef by allowing one system to assign activities for the cooks.

On many days, we are unprepared for cooking items in the kitchen because we did not prepare the required materials properly the day before. We would like a solution that will help us properly prepare the required materials for the next day to improve the efficiency in our kitchen.

Cleaning dishes

Busboys in our restuarant perform two main tasks: clean tables and clean dishes. They have be in both the kitchen and the dining area and it causes a lot of running around. It is also hard for the chef to find a busboy to clean dishes. This decreases the efficiency of the chef because he needs to be in the kitchen to manage the cooks. Some restaurants mitigate this problem by having some busboys stay in the kitchen as dishwashers. However, this is an inefficient solution because the dishwashers are not needed when there are no dishes to be washed and the busboys don't need to be in the dining area when there are no dirty tables. Moreover, this requires us to hire more people for this solution.

We would like to have a system that will tell the busboys when the dishes or tables need to be cleaned. The chef's should be able to contact any of the busboys that are free. Overall we want less clutter in the dining area and kitchen.

Glossary of Terms

Manager - Responsible for managing employees, overseeing the daily operations of the restaurant, and ensuring profitability of the restaurant.

Kitchen - The area where the chef(s) cooks and prepares food. Also, the busboy(s) transports and cleans dishes in this area.

Chef - A professional cook that prepares food for the restaurant.

Customer - A person that purchases food or service at the restaurant.

Waiter/Waitress - Responsible for taking orders and completing requests for customers to ensure customer satisfaction.

Host/Hostess - Manages the restaurant lobby and makes seating arrangements for customers .

Busboy - Cleans dishes and clears tables for the restaurant.

Dining Area - The area where customers eat their meals.

Cashier - A waiter responsible for completing customer transactions.

Split Check - A check that is customized based on all the items purchased by one individual.

Lobby - The area where the customer checks into the restaurant and waits for a table.

Technical Terms

Graphical User Interface - A visual interface that allows workers at the restaurant to interact with the system

Order Queue - lists out food orders to the chef based on first come first serve

Database - A storage device that archives daily restaurant sales, inventory, and feedback.

Floor Plan - An interface that is used to show which tables are currently busy, free ,or dirty.

Notification - A way to alert employees so they can run the restaurant more efficiently.

User Stories

High (core) Priority
Medium Priority
Low Priority

Waiter

Identifier	User Story	Size
ST-W-1	As a waiter, I can remotely send customer orders directly to the chef	6
ST-W-2	As a waiter, I can choose to split the check multiple ways or generate one check.	5
ST-W-3	As a waiter, I can receive notifications when the food is ready for my table.	4
ST-W-4	As a waiter, I can receive notifications when a table requests my assistance.	3
ST-W-5	As a waiter, I can see how long each order will take approximately based on the data collected from previous orders.	5

Customer

Identifier	User Story	Size
ST-C-1	As a customer, I can remotely make a reservation for the restaurant.	4
ST-C-2	As a customer, I can receive remote notifications that indicate when my table is available.	2
ST-C-3	As a customer, I can alert the waiter to request assistance.	3

Host

Identifier	User Story	Size
ST-H-1	As a host I shall be able to see a floor plan of the dining room with which seats are available.	7
ST-H-2	As a host I shall be able to mark tables as reserved or available.	3
ST-H-3	As a host I should be able to see real time updates of the expected time a table should be done.	6

Busboy

Identifier	User Story	Size
ST-B-1	As a busboy I shall be immediately notified when a table needs to be cleaned	3
ST-B-2	As a busboy I shall be notified when I should go to the kitchen to clean dishes	3
ST-B-3	As a busboy I shall be able to mark tables as cleaned	2
ST-B-4	As a busboy I shall be notified when to bring more refreshers to tables.	3

Manager

Identifier	User Story	Size
ST-M-1	As a manager, I can see the status of every item in the inventory. Such as if an item is running low, it will give me an indication.	6
ST-M-2	As a manager, I can modify accounts and permission for each employee and add or remove new ones as necessary.	5
ST-M-3	As a manager, I can view all the transactions that took place in a given day.	3
ST-M-4	As a manager, I can view statistics on the sales in a given day.	4

Chef

Identifier	User Story	Size
ST-CF-1	As a chef, I can view orders that have been placed by the customers	4
ST-CF-2	As a chef, I can notify the waiter that the order is done	4
ST-CF-3	As a chef, I can manage the inventory, and notify the manager when items are running low	7
ST-CF-4	As a chef, I can prioritize and manage orders in the queue	6
ST-CF-5	As a chef, I can view the status of each table	3
ST-CF-6	As a chef, I can alert the busboy to clean the dishes	2
ST-CF-7	As a chef, I can make the use of raw materials more efficient using the data from the system	7
ST-CF-8	As a chef, since I pre-make dishes, I can get information on the most popular dishes of the past day, so I can decide which dishes to prep before tomorrow's opening.	6

Functional Requirements Specification

Stakeholders

Restaurant Owners/Managers - Restaurant owners/Managers would primarily have interest in our system because it provides a method of automation to optimize the efficiency of the restaurant. In turn, the owners would benefit from a rise in profit and a decrease in wasted resources (i.e. time, money, etc).

Customers - Customers benefit from the system in two ways without being the primary users of the system. The customers benefit from the improved service generated by the system itself (i.e. food tracking, instant service requests, etc). Also, the reservation system provides an easy way for customers to make reservations at a particular date and time without having to interact with a person.

Restaurant Employees - Restaurant employees benefit from this system because many of their daily tasks and responsibilities are automated by the system, thus increasing coordination, reducing human error, and reducing wasted resources.

Actors & Goals

Customer
Role: Actor visiting the restaurant to order food and or drinks.
Goal: Customer makes reservations, places orders, and calls waiter.

Waiter
Role: Employee of restaurant that is in charge of taking care of customers.
Goal: Waiter places the order, get notification when food is ready or when the customer requests assistance, and prepares the bill.

Host
Role: Employee that greets customers in the lobby area.
Goal: Uses floor plan to assign tables to customers and manages reservations.

Chef
Role: Employee that prepares all the food for customers in the kitchen
Goal: Gets orders from the waiter and prepares cooking. Notifies waiter when food is ready. Prepares food for the next day based on previous statistics. Notifies Busboy when dishes are full/dirty.

Busboy
Role: Employee that is responsible for keeping the restaurant area clean
Goal: Busboy gets notified when he/she has to clean the tables and dishes.

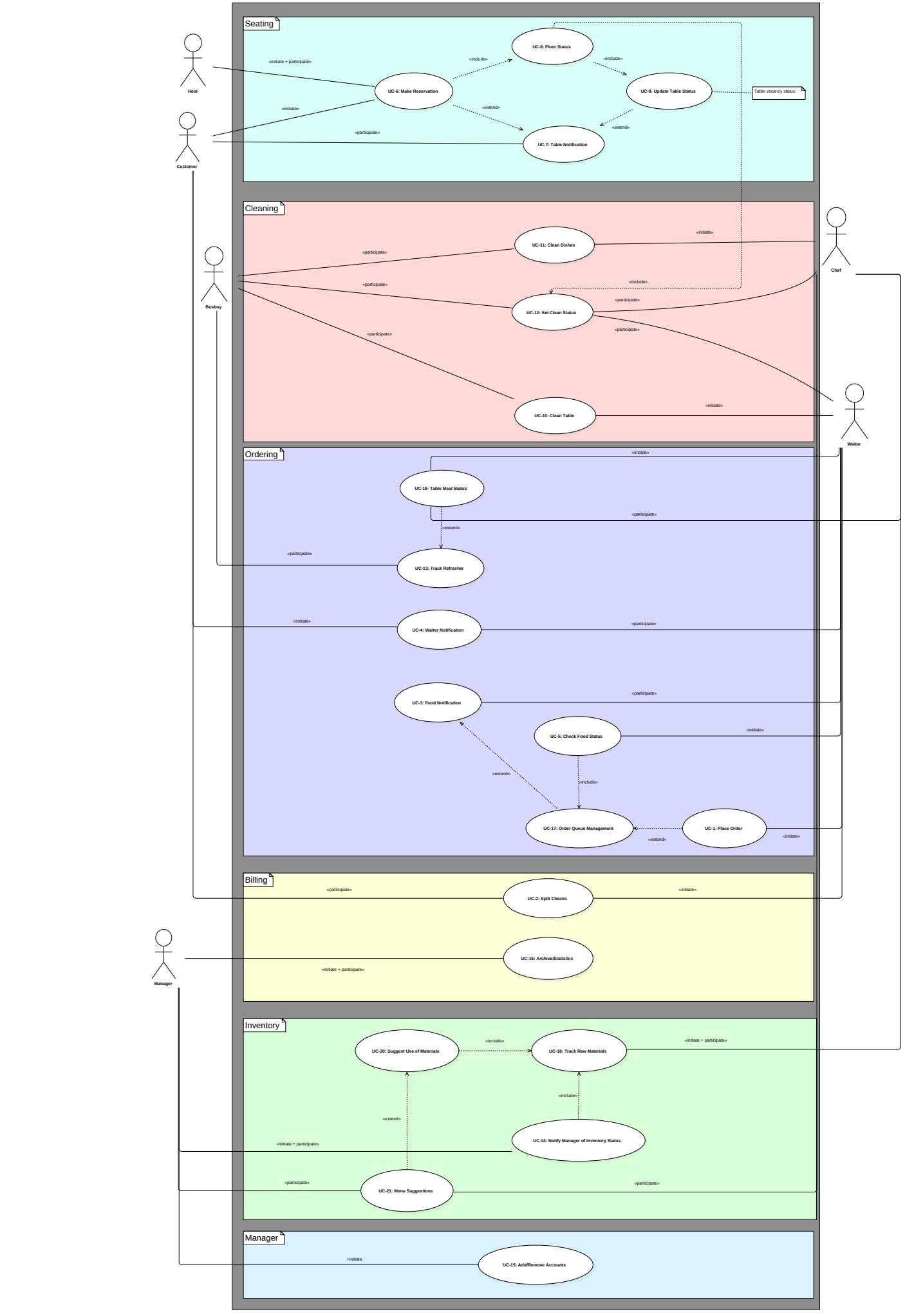
Manager
Role: Employee that oversees every other employee and manages the entire restaurant
Goal: Gather data from restaurant transactions and manage all employee accounts

Use Cases

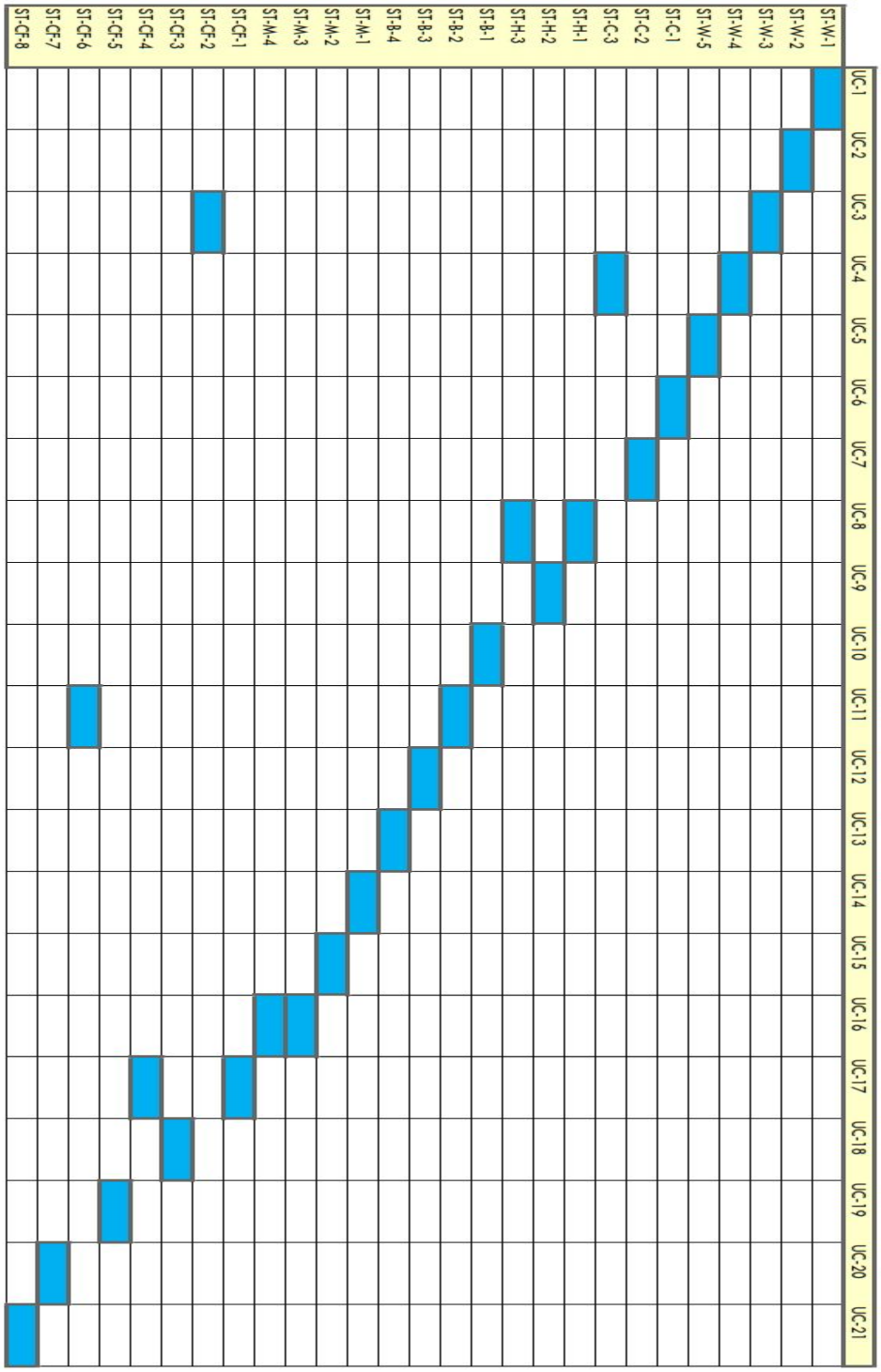
Casual Description

Our group's user stories accommodate the causal description requirement.

Full Use Case Diagram::Full Use Case Diagram



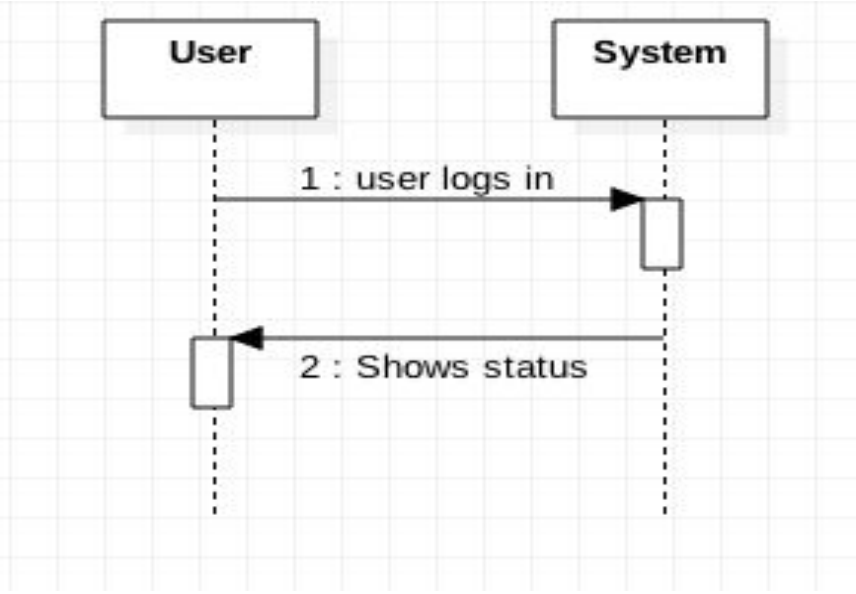
Traceability Matrix



Seating	Ordering	Inventory	Cleaning	Billing	Manager
UC-6 Reservation	UC-1 Place Order	UC-14 Notify Manager of Inventory Status	UC-10 Clean Table	UC-2 Split Checks	UC-15 Add/Remove Accounts
UC-7 Table Notification	UC-3 Food Notification	UC-18 Track Raw Materials	UC-11 Clean Dishes	UC-16 Archive/Statistics	
UC-8 Floor Status	UC-4 Waiter Notification	UC-20 Suggest Use of Materials	UC-12 Set Clean Status		
UC-9 Update Table Status	UC-5 Check Food Status	UC-21 Menu Suggestions			
	UC-13 Track Refresher				
	UC-17 Order Queue Management				
	UC-19 Table Meal Status				

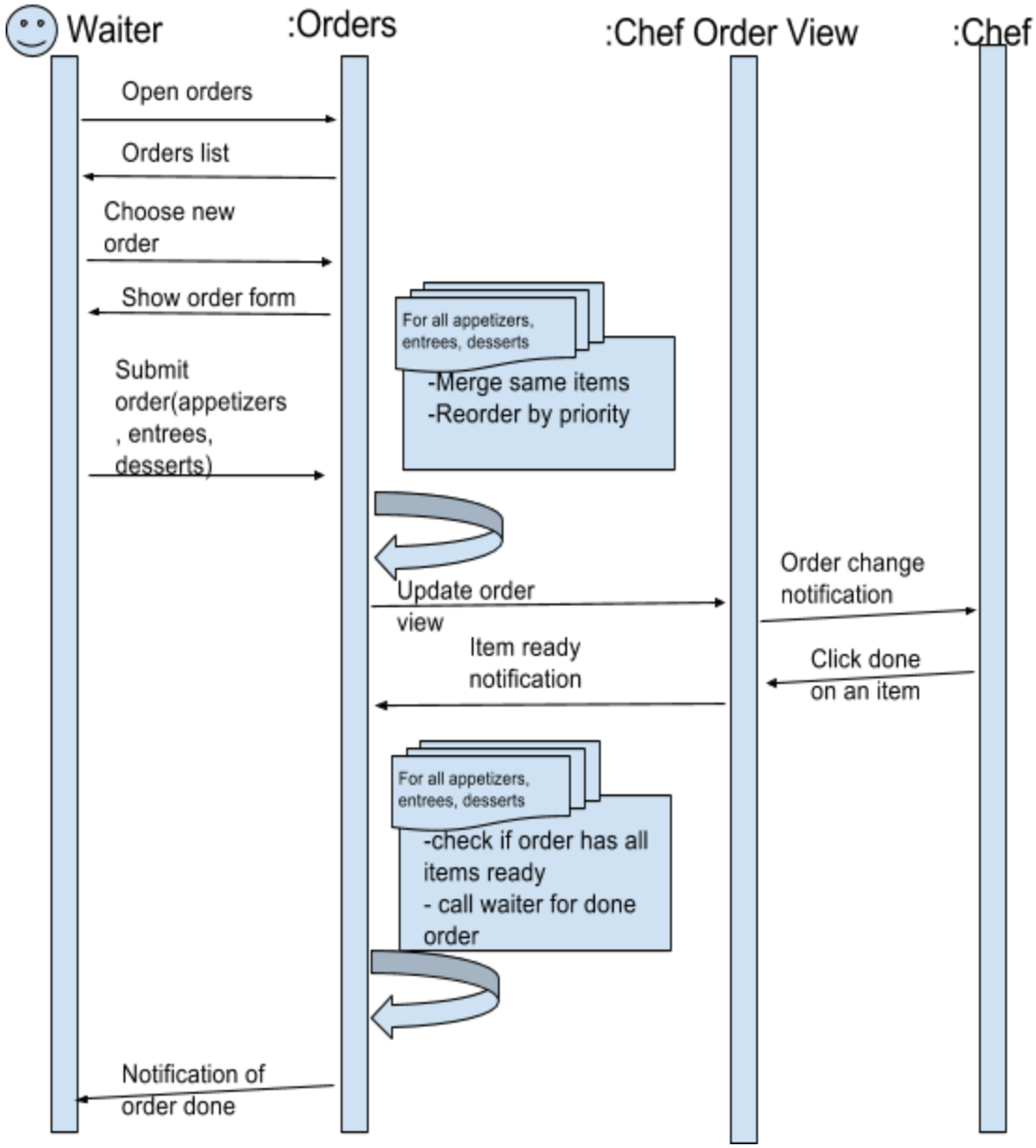
Fully-Dressed Description

Use Case UC-8 Floor Status
Related User Stories: ST-H-1, ST-H-3
Initiating Actor: Host
Actor's Goal: Uses floor plan to assign tables to customers and manages reservations.
Participating Actors: Busboy, Waiter
Preconditions: ● Database already includes Empty, Busy, Dirty and Reserved Status ● Shows the number of seats on each table
Postconditions: ● Each table shows correct status
Flow of Events for Main Success Scenario: -> 1. User logs into system <- 2. System indicates status using different color schemes and patterns



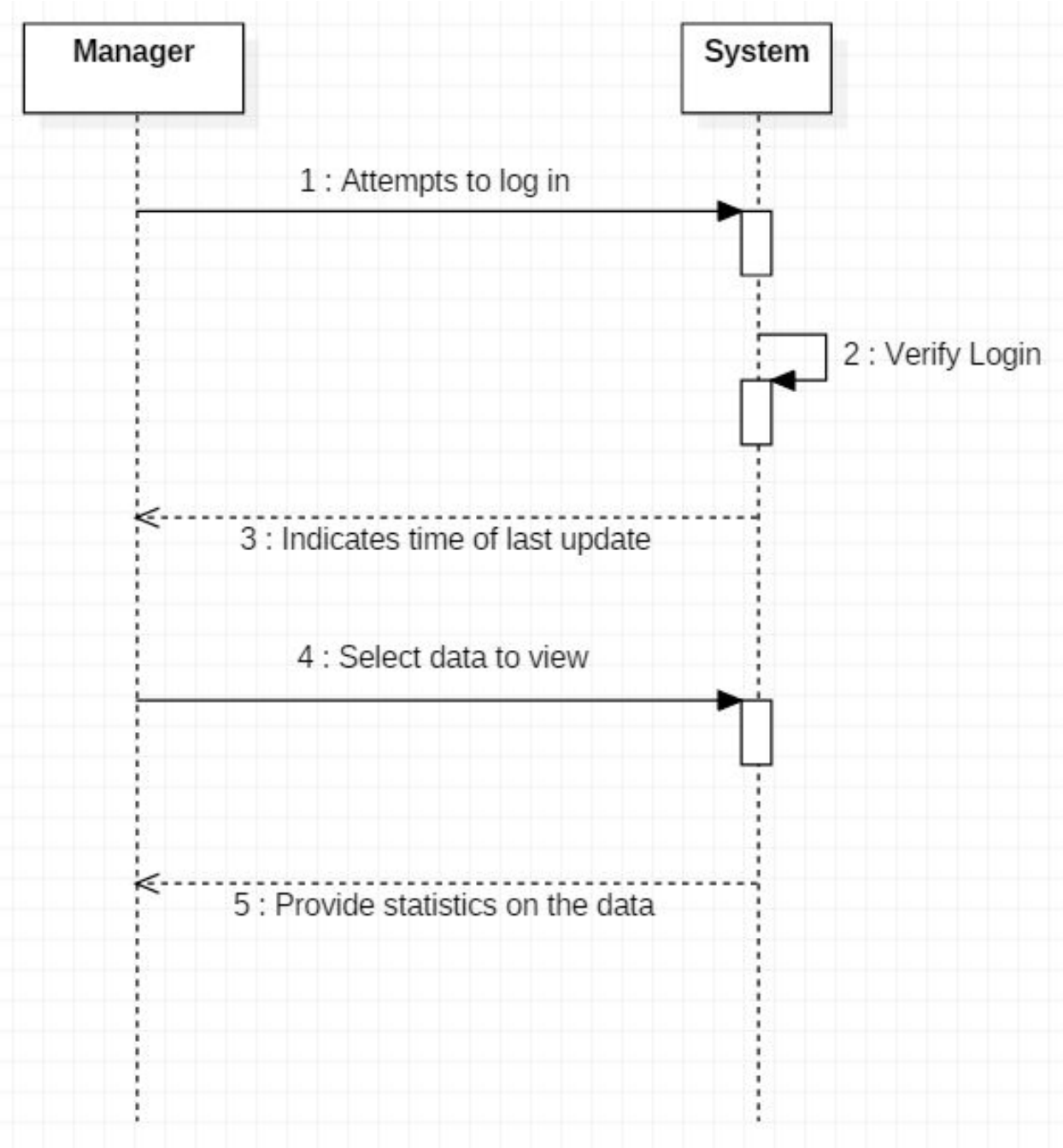
Use Case UC-17 Order Queue Management
Related User Stories: ST-CH-1, ST-CH-4
Initiating Actor: Chef
Actor's Goal: Order placed by waiters are organized into a queue so that they can be distributed fairly to the customers. In addition, it must take into account changes in orders.
Participating Actors: Waiter, Customer
Preconditions: ● Customer told the waiter their order ● Other orders might or might not have been placed
Postconditions: ● The customer gets each of the portions of their order fairly (no one table got all of their portions before another). ● If the customer sent back their order for any reason it will eventually be distributed to them correctly
Flow of Events for Main Success Scenario: 1) Waiter inputs order to system 2*) Order is organized into queue system based on what portion of the whole meal it takes up (i.e. appetizers before main meal). 2*) Order was possibly sent back for some reason and reorganized back into the queue 3) Waiter is eventually notified of order.

Sequence Diagram Use case UC-17



Use Case UC-16 Archiving/Statistics
Related User Stories: ST-M-3, ST-M-4
Initiating Actor: Manager
Actor's Goal: Be in charge of daily sales and transaction of the restaurant.
Participating Actors: None
Preconditions: ● There was at least one customer who visited and paid for the bill ● The database system was updated after the restaurant was closed
Postconditions: ● Each statistics gives accurate sales, averages, and sums
Flow of Events for Main Success Scenario: -> 1. Manager logs into system <- 2. System verifies login <- 3. System indicates time of last update -> 4. Manager selects data he/she wants to view <- 5. System provides statistics on select option

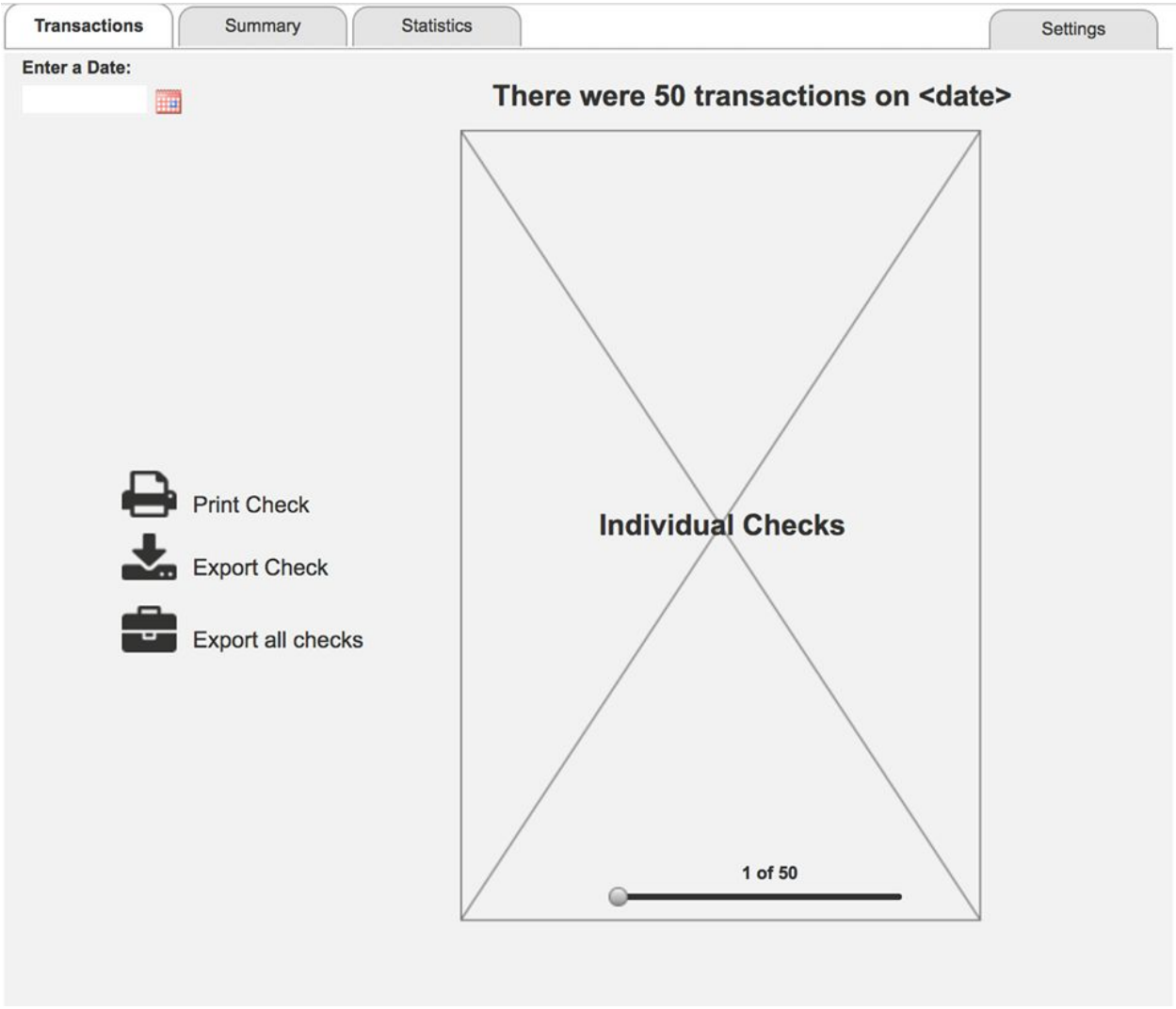
Sequence Diagram: Use Case UC-16



User Interface Specification

Preliminary Design

The manager will be able to see all the checks for the given date, and be able to export them if he needs to. From this page the manager can navigate to the summary or statistics tab to view more information.



Transactions | **Summary** | Statistics | Settings

Start Date: **End Date:**

-

Total sales : $XXXX.XXXX

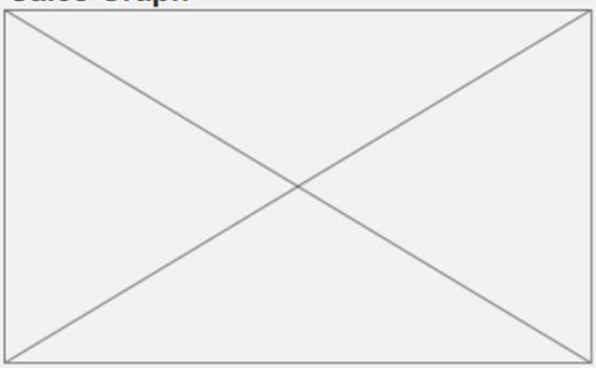
Total profit: $XXXX.XXXX

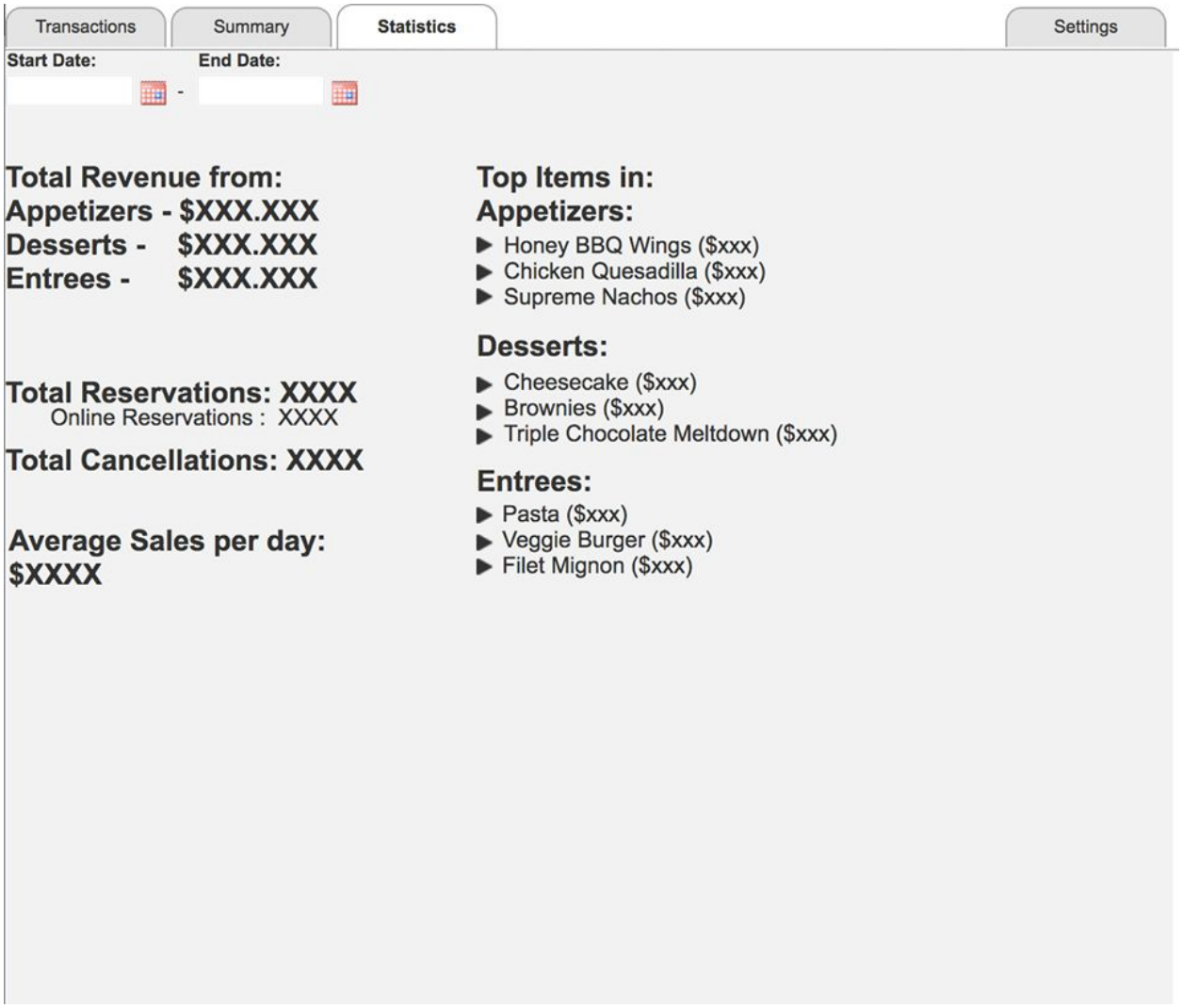
Total checks: XXX

Total items sold: XXX

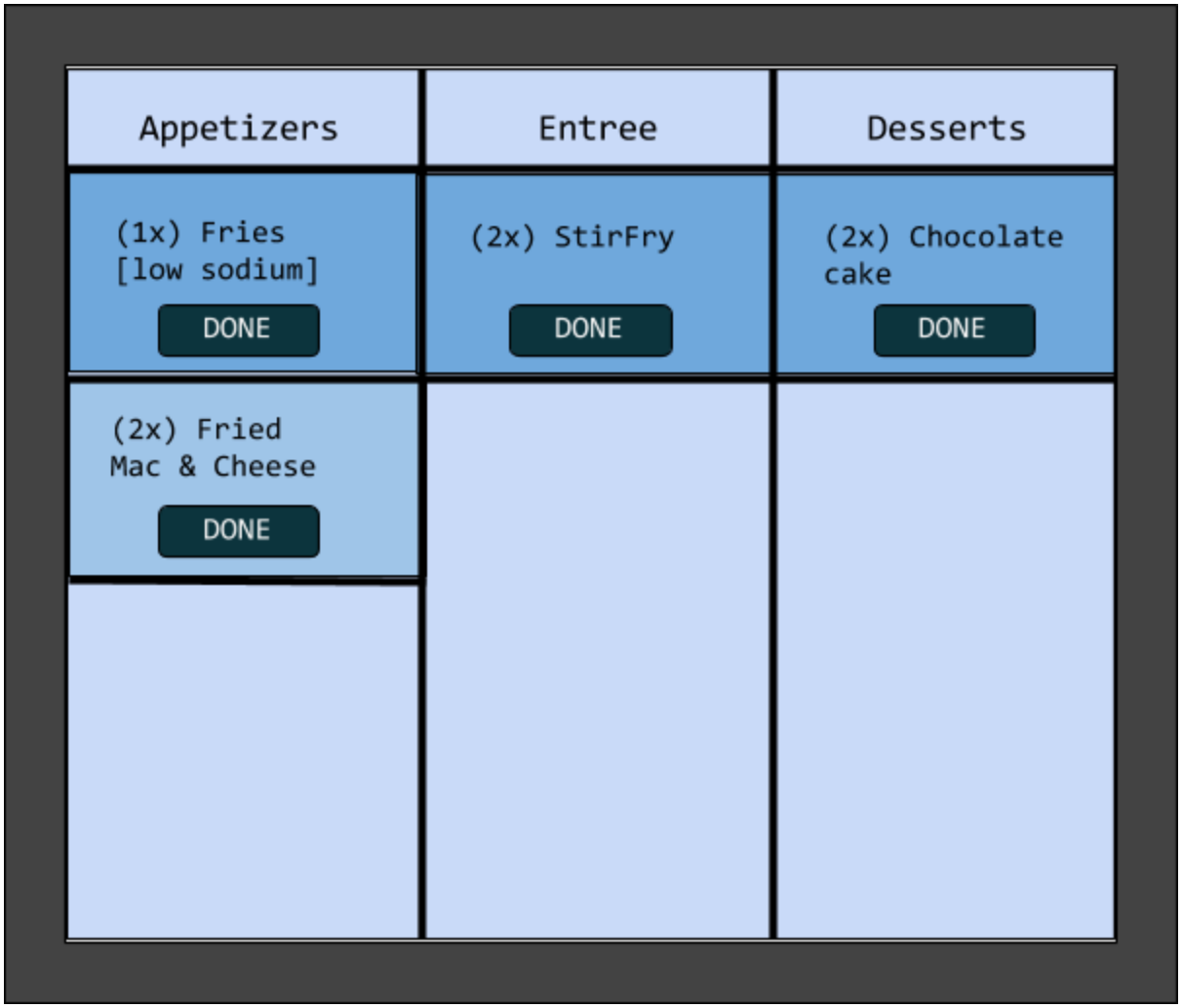
Most Popular Item:
Appetizers: Calamari
Desserts: Tiramisu
Entrees: Chicken Parmesan

Sales Graph

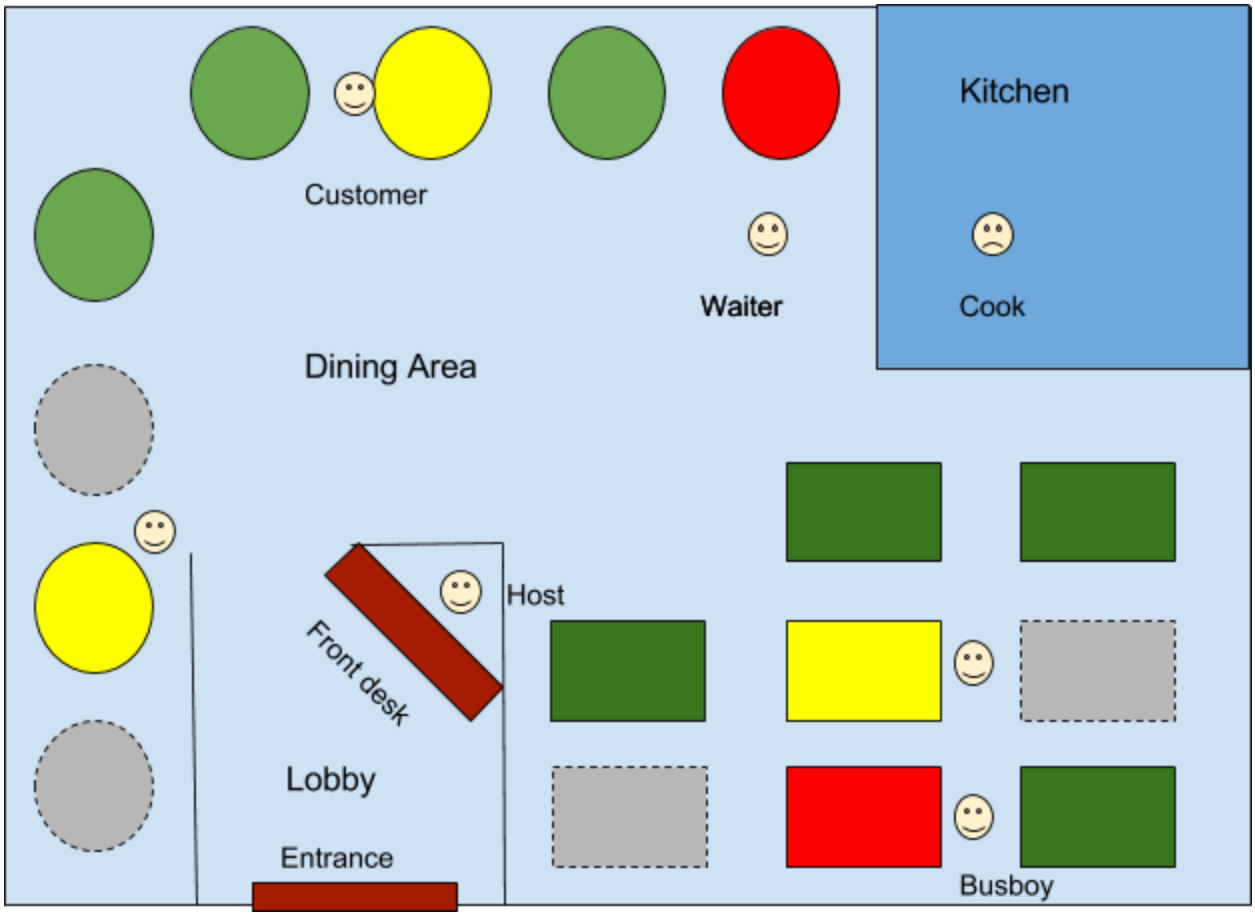



Transactions
Summary
Statistics
Settings
Start Date:
End Date:
-
Total Revenue from:
Appetizers - $XXX.XXX
Desserts - $XXX.XXX
Entrees - $XXX.XXX
Total Reservations: XXXX
Online Reservations : XXXX
Total Cancellations: XXXX
Average Sales per day:
$XXXX
Top Items in:
Appetizers:
Honey BBQ Wings ($xxx)
Chicken Quesadilla ($xxx)
Supreme Nachos ($xxx)
Desserts:
Cheesecake ($xxx)
Brownies ($xxx)
Triple Chocolate Meltdown ($xxx)
Entrees:
Pasta ($xxx)
Veggie Burger ($xxx)
Filet Mignon ($xxx)

The Chef will be able to see which items on the list he/she should get done first. The chef could also modify the queue if needed.



Markup Floor Plan



Legend:

- Available Table
- Occupied Table
- Dirty Table
- Reserved Table

User Effort Estimation

Scenario 1: Floor Status

1. User logs in and can view the whole floor plan
    a. If user wants more information on a specific table, they can tap on it
    b. Tap on back after user is done with looking at information

Estimated Total Clicks: 2-4

Scenario 2: Order Queue Management

1. User logs into system
2. Waiter navigates around categories to place customer orders
3. Chef navigates order around based on priority
4. Chef hits done when food is ready
5. Waiter confirms notification through the device

Estimated total clicks 5-15

Scenario 3: Archiving/Statistics

1. User logs into system
2. User navigates to archive
    a. If user wants to view transaction tab they just click once on it
    b. If user wants to view summary tab they just click once on it
    c. If user wants to view statistics tab they just click once on it
3. User goes back/logs out of system

Estimated total clicks: 3-6

All team members managed the project equally!

References

StarUML

http://stackoverflow.com/questions/1696927/whats-is-the-difference-between-include-and-extend-in-use-case-diagram

http://www.ece.rutgers.edu/~marsic/Teaching/SE/report1.html

Axure