

# RESTAURANT INFORMATION SYSTEM

## PART 1

## REQUIREMENTS (1)

### PRIMARY STAKEHOLDERS

- Host
- Wait Staff
- Kitchen Staff
- Bus Boys
- Management
- Customers

### STAKEHOLDERS' REQUIREMENTS (1A)

#### ALL STAFF

- As a staff member, I want to be able to **log in and out of the system** so that I can access its information and functionality.
- As a staff member, I want to be able to **reset my password** so that I will be able to access the system even if I forget my password.
- As a staff member, I want to be able to **report a mess** so that busboys will be able to clean it up.

#### HOST

- As a host, I want to be able to **view the status of tables** (including which wait staff and bus boys are manning which tables) so that I can coordinate with the waiters and effectively find tables for customers.
- As a host, I want to be able to **mark a table as occupied when customers are seated** so that I can keep track of which tables are available and so that the waiters are updated when a table becomes occupied.
- As a host, I want to **get an estimate of the wait time for a party of customers** so that I can provide good service.
- As a host, I want to be able to **record the name, phone number, and number of customers in a party for customers waiting for seating** so that they can be notified when a table is ready.

#### WAIT STAFF

- As a waiter, I want to be to **enter a customer's order** so that it can be delivered to the kitchen and filled.
- As a waiter, I want to **receive notifications when customers' orders are prepared** so that I can coordinate with kitchen staff and deliver orders to customers promptly.
- As a waiter, I want to **mark table as unoccupied when customers leave** so that the bus boys and host can make sure that tables are promptly cleaned and made available for customers.

- As a waiter, I want to **modify customers' orders (send back or add items) if necessary** so that I can provide good service to customers.
- As a waiter, I want to **add notes to customers' orders** so the kitchen staff can make changes to a dish as necessary and so that customers get the dish they want.
- As a waiter, I want to be able to **select a table and enter payment for that table's current order** so that I can collect payments.

## KITCHEN STAFF (COOKS)

- As a cook, I want to **see all pending orders in order of when they were submitted** so I can make sure customers don't wait too long to get their food.
- As a cook, I want the system to **mark orders as complete** so that the system can notify the waiters when it's ready and the food can be delivered quickly.
- As a cook, I want to **receive notifications when an order is sent back with notes about what to fix** so that the restaurant provides good service to unsatisfied customers.
- As a cook, I want to be **notified when customers add items to an order** so I can prepare them promptly.

## BUSBOYS

- As a busboy, I want to **receive a notification when customers leave a table** so that I know what tables need to be cleaned.
- As a busboy, I want to **mark a table as cleaned once clean** so that wait staff and the host are notified once a table is clean.
- As a busboy, I want to **receive notifications about messes/spills** so that they get cleaned up promptly.

## MANAGEMENT

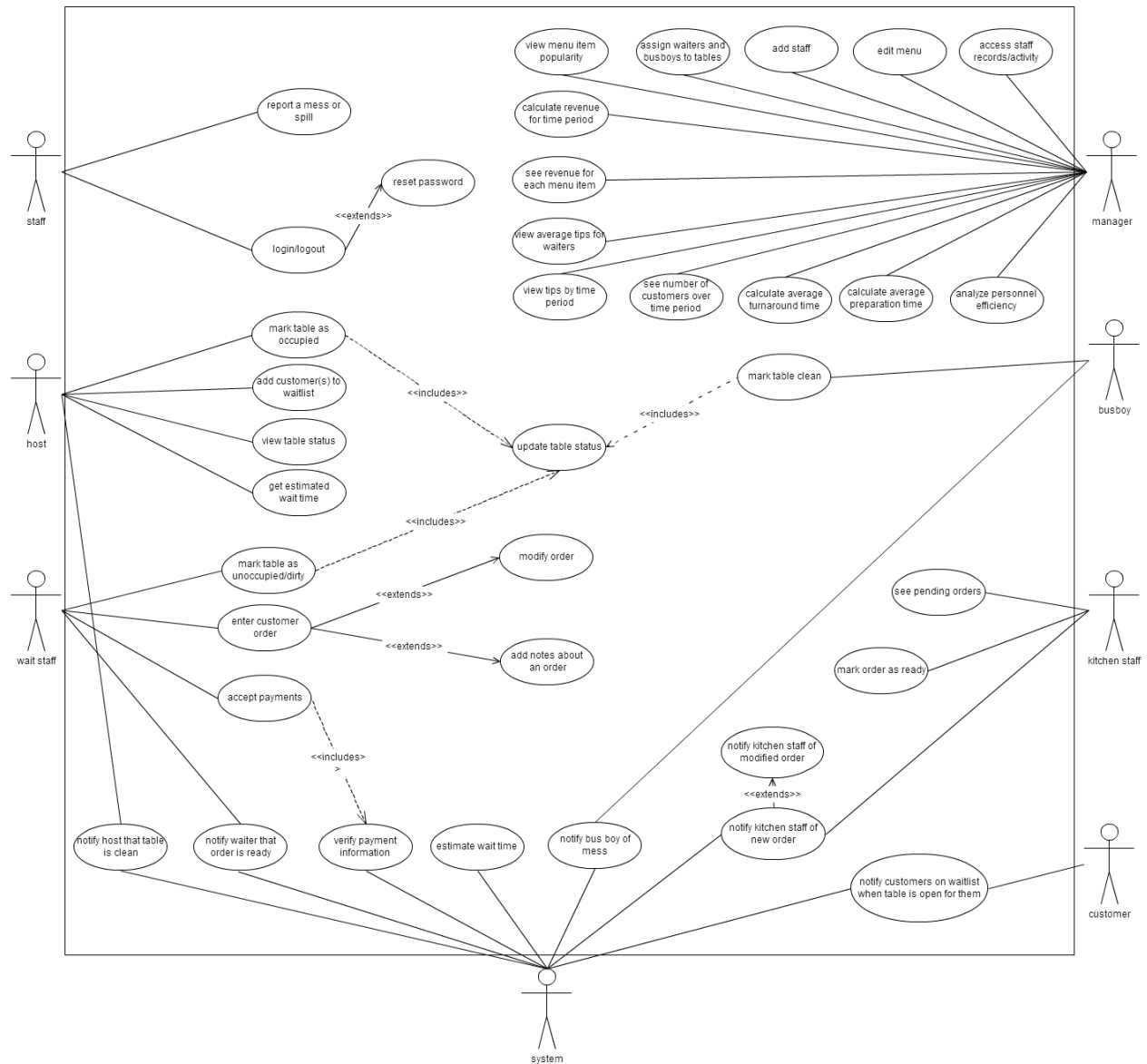
- As a manager, I want to **access staff records/activity in the system** so that I can use that information to handle payroll and to improve service.
- As a manager, I want to **edit (add, remove, change) items from the menu** so that our offerings can change over time and increase profits.
- As a manager, I want to **add new staff members to the system** and send them login credentials so that new employees can also use the system.
- As a manager, I want to **assign (and reassign) waiters and busboys to tables** so that staff's activity is well coordinated and efficient, and so that the system can track staff's work.
- As a manager, I want to **see data about what menu items are most popular** so that we can better meet customers' needs and grow our business.
- As a manager, I want to **see data about personnel efficiency** so that I can use that information to inform payroll decisions and to better coordinate staff activities.
- As a manager, I want to **see data about the average turnaround time** (how long patrons spend in the restaurant) for orders so that we can improve our services and identify any problems causing delays.

- As a manager, I want to **see data about the average preparation time** (time from when the order is placed to when it is ready) for orders so that we can improve our services and identify any problems causing delays.
- As a manager, I want to **see the number of customers who come to the restaurant throughout the day** so that I can see spikes in business and better coordinate staff so that we can handle those spikes.
- As a manager, I want the system to **calculate the revenue for a given time period** (day, week, month, year, etc.) so that I know how my business is doing and can adapt accordingly.
- As a manager, I want the system to **show me the revenue associated with each menu item** so that we can understand which parts of our menu best serve customers' needs.
- As a manager, I want to be able to **see the tips collected for a given time period** so that I can understand how our service is doing at different times.
- As a manager, I want to be able to **see the average tips collected by each staff person** so that I know who is providing the best service to customers.

## CUSTOMER

- As a customer, I want to **receive a text message when my table is ready** so that I know when it's ready (whether or not I'm in the restaurant at the time).

## USE-CASE DIAGRAM (1B)



See file "restaurant\_use\_case\_diagram.jpeg" for larger version of this diagram.

## STAKEHOLDERS' QUALITY ATTRIBUTE REQUIREMENTS (1C)

### ALL STAFF

**Usability:** It should be easy to login and logout. Logging in and out of the system should be simple and quick. Logins should last for staff's entire shift when applicable. Logout should be obvious so that users will not forget to logout of a shared device.

### HOST

**Performance:** There should be real-time updates for tables with a maximum of 30 seconds of lag time. This is important because all the staff members need the most up to date information in order to seat and serve their customers as efficiently as possible.

**Usability:** It should be easy to see status of all tables at once. Not being able to see all tables at once would make it difficult for the host to find seats for new customers at a glance, which would likely decrease performance. Also, as the current whiteboard system allows for this capability, removing it would likely frustrate users.

### WAIT STAFF

**Performance:** The order should be sent to the kitchen quickly. There should be real-time updates for when the order is placed by the wait staff to when the order appears for the kitchen staff with a maximum of 30 seconds of lag time. A high performance level will ensure that the system speeds up (rather than delaying) the preparation of customers' orders.

**Usability:** It should be quick to enter orders. Waiters are accustomed to taking orders down on paper in shorthand, and the speed of entry on this system should aim to match or better the speed of written orders.

**Usability:** The application should be designed to prevent errors. Since waiters have to take customers' orders down quickly, and may be stressed during peak periods, the applications' design should prevent errors in order entry (e.g., using easily readable and large interface elements).

### KITCHEN STAFF

**Performance:** Orders should be received quickly. Kitchen staff use order information to determine the work they do, so it is important that the system not delay their work.

**Usability:** The display should have minimal hand-use so that kitchen staff does not have to set down cooking implements or contaminate their gloves while preparing food.

**Usability:** The display should be readable from a distance. Staff should be able to see a new order without having to walk across the kitchen so that the process of checking orders doesn't interrupt their more important work of preparing food.

**Robustness:** It should work in the environment of a kitchen— e.g., being heat resistant, water resistant, and easy to clean. The system needs to function well in and withstand a kitchen environment.

## BUS BOYS

**Performance:** Bus boys should be notified of dirty tables and spills quickly so that they do not compromise customers' experience or impede waiters.

## CUSTOMERS

**Security:** Credit card information should be stored safely so that private financial information is not compromised.

**Performance/Usability** Waiters should be able to enter orders quickly and accurately so that I don't have to wait while they finish entering an order.

## MANAGEMENT

**Security:** Unauthorized people should not be able to access the system, as it contains potentially sensitive business data. (*Security*)

**Availability/Reliability:** The system downtime should be minimal; it should not break down during working hours and maintenance should be scheduled consistently. The system is integral to the restaurant staff and managements' workflow, so its downtime needs to work around those users' work times.

**Availability/Reliability:** Data should be safely backed-up. The system data (e.g., restaurant's performance over time, finance, staff records) is essential to the business and should be recoverable in the case of a system defect.

**Usability:** It should be easy to index business data on various factors so that the system supports management's analysis and future planning. (*Usability*)

## QUALITY ATTRIBUTE SCENARIOS (1D)

### QAS1: PERFORMANCE OF TABLE UPDATES

<b>Source of Stimulus</b>	Busboy
<b>Stimulus</b>	Reports that a dirty table is now clean/available
<b>Artifact</b>	The Restaurant System
<b>Environment</b>	Under Normal Conditions
<b>Response</b>	The hosts' table display is updated and the system sends a notification to the designated waiter
<b>Response Measures</b>	Within 30 seconds of the waiter's report

### QAS2: SECURITY OF SYSTEM OVERALL

<b>Source of Stimulus</b>	An unauthorized individual
<b>Stimulus</b>	Tries to access customer payment information
<b>Artifact</b>	The Restaurant System
<b>Environment</b>	While connected to the restaurant's Wi-Fi
<b>Response</b>	Does not allow the user to log in or access information and stores customer payment information in an encrypted format
<b>Response Measures</b>	Such that it would require years to decrypt and access the information with a standard laptop computer.

### QAS3: USABILITY FOR KITCHEN STAFF VIEWING AND UPDATING ORDERS

<b>Source of Stimulus</b>	Kitchen Staff
<b>Stimulus</b>	Wishes to feel comfortable while using
<b>Artifact</b>	The Restaurant System
<b>Environment</b>	At runtime (in the kitchen, while cooking)
<b>Response</b>	System displays the state in an easy to read format and has buttons that are usable for someone wearing gloves
<b>Response Measures</b>	Such that kitchen staff can view and update orders in less than 3 seconds



#### QAS4: USABILITY FOR WAIT STAFF ENTERING ORDERS

<b>Source of Stimulus</b>	Waiter
<b>Stimulus</b>	Wants to quickly and efficiently enter a customer's order
<b>Artifact</b>	The Restaurant System
<b>Environment</b>	At runtime
<b>Response</b>	Should let waiters reuse previously entered data about orders and autocomplete orders or notes when appropriate
<b>Response Measures</b>	Such that waiters can enter an order in real-time as customers provide it, and such that 95% of wait staff are satisfied with the system (when surveyed)

## RANKED USER STORIES (1E)

### A LEVEL (MUST HAVE)

#### ALL STAFF

- As a staff member, I want to be able to **log in and out of the system** so that I can access its information and functionality.

#### HOST

- As a host, I want to be able to **view the status of tables** so that I can coordinate with the waiters and effectively find tables for customers.
- As a host, I want to be able to **mark a table as occupied when customers are seated** so that I can keep track of which tables are available and so that the waiters are updated when a table becomes occupied.

#### WAIT STAFF

- As a waiter, I want to be to **enter a customer's order** so that it can be delivered to the kitchen and filled.
- As a waiter, I want to **receive notifications when customers' orders are prepared** so that I can coordinate with kitchen staff and deliver orders to customers promptly.
- As a waiter, I want **to modify customers' orders (send back or add items) if necessary** so that I can provide good service to customers.
- As a waiter, I want **to add notes to customers' orders** so the kitchen staff can make changes to a dish as necessary and so that customers get the dish they.

#### KITCHEN STAFF

- As a cook, I want to **see all pending orders in order of when they were submitted** so I can make sure customers don't wait too long to get their food.
- As a cook, I want the system to **mark orders as complete** so that the system can notify the waiters when it's ready and the food can be delivered quickly.
- As a cook, I want to be **notified when customers add items to an order** so I can prepare them promptly.

#### BUS BOYS

- As a busboy, I want to **mark a table as cleaned once clean** so that wait staff and the host are notified once a table is clean.

#### MANAGEMENT

- As a manager, I want to **access staff records/activity in the system** so that I can use that information to handle payroll and to improve service.
- As a manager, I want to **edit (add, remove, change) items from the menu** so that our offerings can change over time and increase profits.
- As a manager, I want to **add new staff members to the system** and send them login credentials so that new employees can also use the system.
- As a manager, I want the system to **calculate the revenue for a given time period** (day, week, month, year, etc.) so that I know how my business is doing and can adapt accordingly.

## B LEVEL (SHOULD HAVE)

### ALL STAFF

- As a staff member, I want to be able to **reset my password** so that I will be able to access the system even if I forget my password.

### HOST

- As a host, I want to be able to record the name, phone number, and number of customers in a party for customers waiting for seating so that they can be notified when a table is ready.

### WAIT STAFF

- As a waiter, I want to **mark table as unoccupied when customers leave** so that the bus boys and host can make sure that tables are promptly cleaned and made available for customers.
- As a waiter, I want to be able to **select a table and enter payment for that table's current order** so that I can collect payments.

### KITCHEN STAFF

- As a cook, I want to **receive notifications when an order is sent back with notes about what to fix** so that the restaurant provides good service to unsatisfied customers.

### BUS BOYS

- As a busboy, I want to **receive a notification when customers leave a table** so that I know what tables need to be cleaned.

### MANAGEMENT

- As a manager, I want to **assign (and reassign) waiters and busboys to tables** so that staff's activity is well coordinated and efficient, and so that the system can track staff's work.
- As a manager, I want to **see data about what menu items are most popular** so that we can better meet customers' needs and grow our business.
- As a manager, I want to **see data about personnel efficiency** so that I can use that information to inform payroll decisions and to better coordinate staff activities.

- As a manager, I want to **see data about the average turnaround time** (how long patrons spend in the restaurant) for orders so that we can improve our services and identify any problems causing delays.
- As a manager, I want to **see data about the average preparation time** (time from when the order is placed to when it is ready) for orders so that we can improve our services and identify any problems causing delays.
- As a manager, I want to **see the number of customers who come to the restaurant throughout the day** so that I can see spikes in business and better coordinate staff so that we can handle those spikes.
- As a manager, I want the system to **show me the revenue associated with each menu item** so that we can understand which parts of our menu best serve customers' needs.
- As a manager, I want to be able to **see the tips collected for a given time period** so that I can understand how our service is doing at different times.

## C LEVEL (COULD HAVE)

### ALL STAFF

- As a staff member, I want to be able to **report a mess** so that busboys will be able to clean it up.

### BUS BOYS

- As a busboy, I want to **receive notifications about messes/spills** so that they get cleaned up promptly.

### HOST

- As a host, I want to **get an estimate of the wait time for a party of customers** so that I can provide good service.

### MANAGEMENT

- As a manager, I want to **see data about the average turnaround time** (how long patrons spend in the restaurant) for orders so that we can improve our services and identify any problems causing delays.
- As a manager, I want to be able to **see the average tips collected by each staff person** so that I know who is providing the best service to customers.

### CUSTOMER

- As a customer, I want to **receive a text message when my table is ready** so that I know when it's ready (whether or not I'm in the restaurant at the time).

## COMPETITIVE ANALYSIS (2)

### SYSTEM A: ACCUPOS ANDROID POINT OF SALE SOFTWARE

#### DESCRIPTION

AccuPos is a general solution point of sale (POS) system. The company sells a specific "Restaurant Bundle" version of their software that can be configured for different types of restaurants (e.g., "Yogurt Shop", "Fast Food"). To use the system, waiters type orders into an Android tablet. Receipts for orders go to a printer in the kitchen, and kitchen staff and wait staff use the receipts to fill and collect payment for the orders. This system (as far as we can tell from information available on the website) stores data in the cloud/on the company's servers.

<http://www.accupos.com/>

#### COST

The hardware base package for the AccuPOS system costs about \$1000. For each Android device, aside from the purchase cost (~\$150 for an Android phone), an additional software license costs \$150. There is an optional fee for a service package of \$25 per month per device for setup, support, training, and upgrades.

#### SCREENSHOTS

Edit or Delete an Item Slide 16 of 18



Menu Example Slide 10 of 18



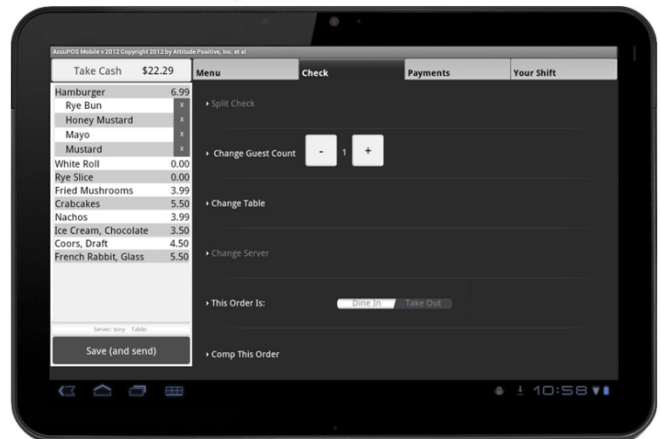
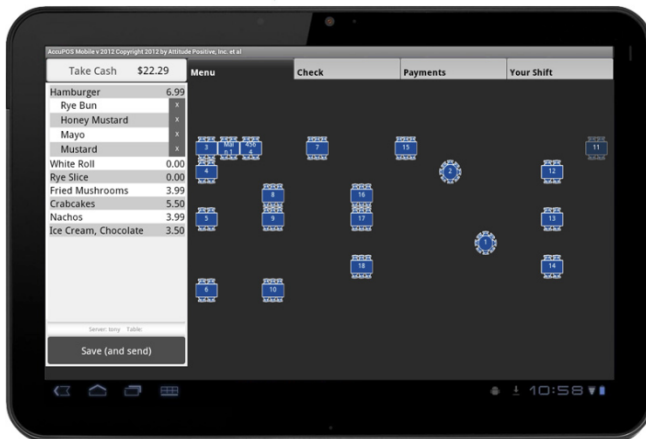


The Tables Layout

Slide 7 of 18

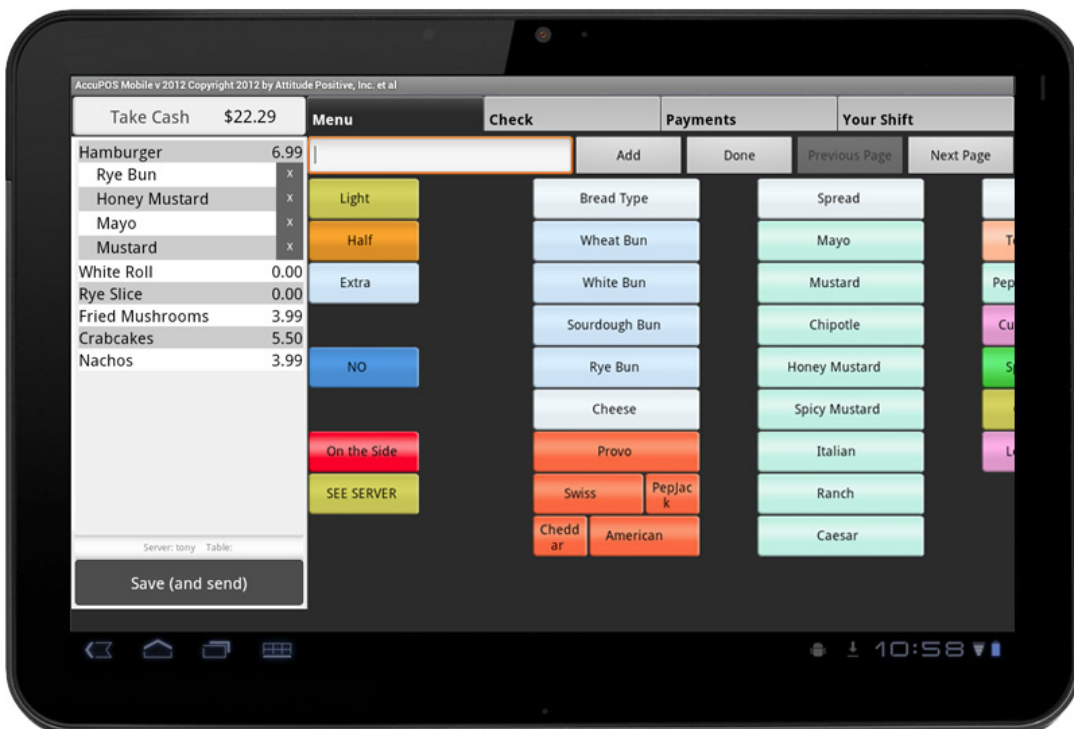
Editing A Check

Slide 6 of 18



Choices for a Sandwich

Slide 3 of 18



## SYSTEM B: REVEL

### DESCRIPTION

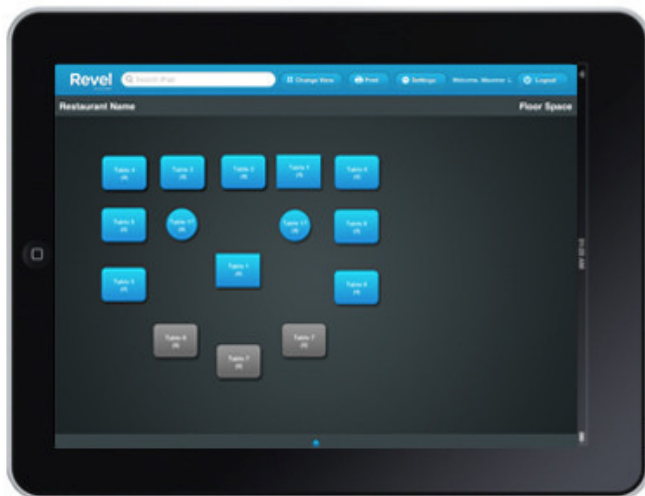
Revel is an iPad application made for specifically for restaurants. It handles orders for waiters and kitchen staff, helps hosts to manage table availability, takes payments, and does employee time tracking. Restaurants can either have waiters use one iPad at a central station or provide iPads for each waiter. Kitchen staff also needs an iPad to see the order queue.

<http://revelsystems.com/>

### COST

The one-time cost for the software is \$1500. The cost for the basic hardware bundle (one iPad, a stand, a receipt printer, etc.) is \$2300. Restaurants can add additional iPads for the cost of the iPad (about \$400 for an iPad 2) and an additional software licensing fee (this might be \$1500, but it is unclear from the information available on the company's website).

### SCREENSHOTS



## COMPARISON CHART

	AccuPos	Revel Systems	Our system
<b>Cost of software</b>	\$795	\$1500	(unknown)
<b>Cost of hardware</b>	\$2299	\$2300	(n/a, we don't provide hardware)
<b>Touchscreen</b>	yes	yes	yes
<b>Mobile</b>	Android tablets and phones	iPad only	any device with a browser
<b>Splashproof</b>	yes		(n/a, we don't provide hardware)
<b>Remote Kitchen Printer (for orders)</b>	yes		
<b>Time Clock for Employee Shifts</b>	yes		yes
<b>Track Employee Tips</b>		yes	yes
<b>Track Register Cash Flow</b>		yes	(unknown)
<b>Installation Included</b>	yes		(unknown)
<b>Training Included</b>	yes		(unknown)
<b>Software Upgrades Included</b>	yes	yes	yes
<b>Email/Telephone Support</b>	yes		(unknown)
<b>Track Tables</b>	yes	yes	yes
<b>Split/Merge Bills</b>		yes	(unknown)
<b>Accounting Information Management</b>	yes	yes	yes
<b>Online Order Handling</b>		yes	
<b>Easily Modify Order</b>		yes	yes
<b>Menu Search</b>		yes	(unknown)
<b>Tablesides Ordering (for customers)</b>		yes	
<b>Customer Database</b>	yes	yes	yes
<b>Cash Drawer</b>	yes	yes	(unknown)
<b>Card Reader</b>	yes	yes	(unknown)
<b>PCI Compliant</b>	yes	yes	yes
<b>Text Message Customers on Waitlist</b>			yes
<b>Automatic notifications for staff coordination</b>			yes
<b>Restaurant analytics dashboard for management</b>			yes



## OUR SYSTEM (1B)

### DESCRIPTION

Our system helps small restaurants manage their daily operations. By managing restaurant operations data and automatically sending information to various staff members, the system improves coordination between the host, the wait staff, the kitchen staff, and the busboys. In addition to supporting restaurant operations, the system gathers data about the menu, customers, and restaurant performance (including turnaround time, preparation time, and the wait-time). The system also tracks employees' wages, hours worked, and tipped so that it can streamline accounting and payroll operations. Moreover, our system provides an online dashboard where restaurant managers can analyze the data the system gathers. The manager can use this information to improve overall efficiency of the restaurant, recognize trends, and discover patterns in customer traffic.

## WHAT MAKES OUR SYSTEM DIFFERENT

Our system differs from existing systems because it is web-based, provides unique functionality for waitlist management, and has a usable dashboard for management to analyze restaurant data.

### WEB-BASED

Competitor point-of-sale systems for restaurants tend to be native applications. Our system is different because it is web-based. The key benefit of the system being web-based is that it allows restaurants to purchase hardware separately from our software, since our application can be accessed by any device with a web browser. We believe that the ability to use our software on any internet-enabled hardware will appeal to small restaurants because it will allow them to have more control over the costs and type of hardware they want to use. In the end, making our system web-based should make it appeal to more of our target audience.

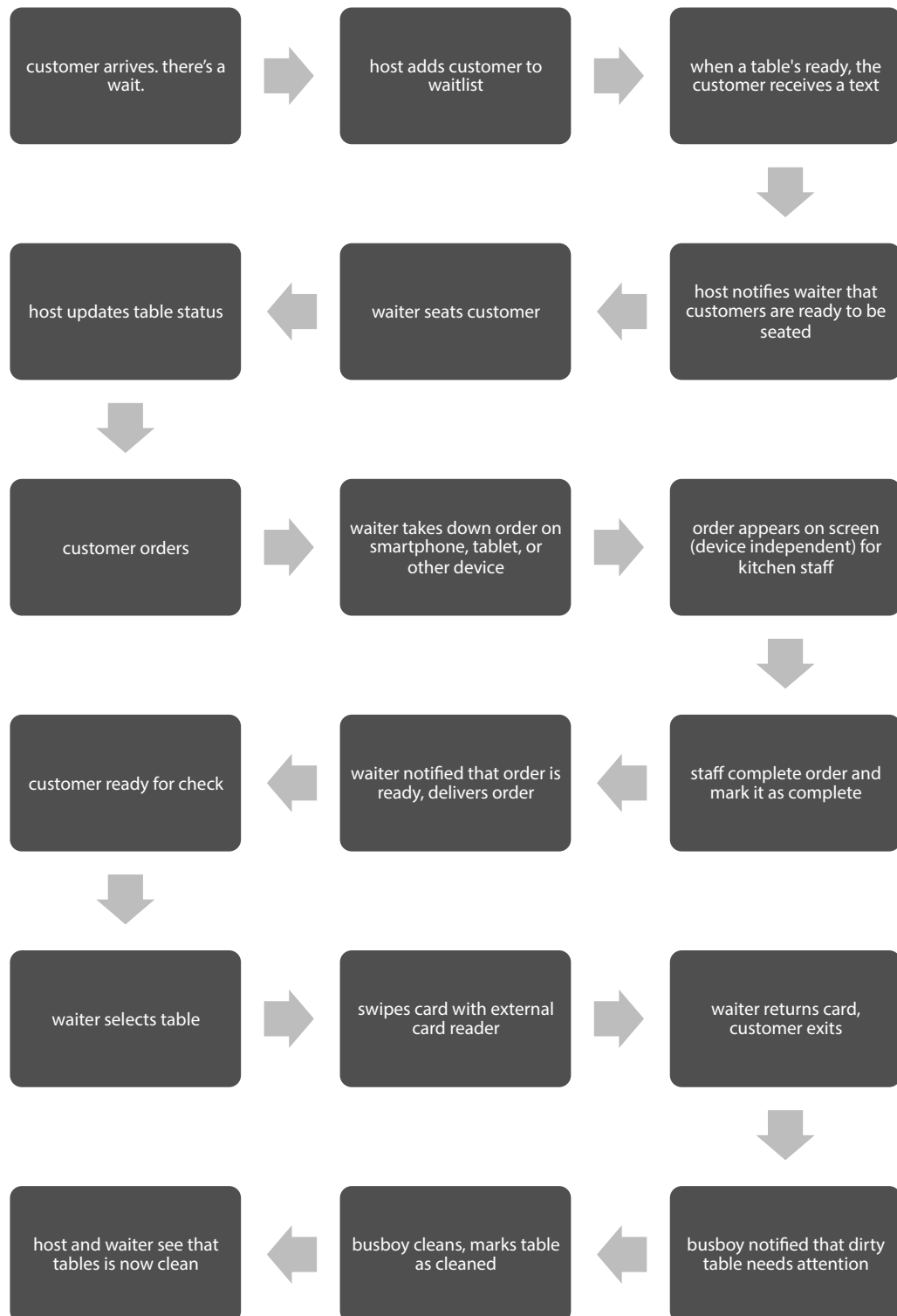
### UNIQUE APPROACH TO WAITLIST MANAGEMENT

Our system adds useful and unique functionality several stakeholders. First, the system provides the capability for hosts to send customers text-message notifications when their table is ready. This takes advantage of widespread cell phone use, and allows the restaurant to handle its waitlist more effectively without having to purchase specialized “buzzer” devices. The system also helps hosts manage wait lists by giving them more accurate estimations of wait time (rather than the host having to guess and irritate customers with inaccurate estimates).

### MANAGEMENT DASHBOARD

The final feature which differentiates our system is its usable dashboard for restaurant managers to handle restaurant data. Many systems collect information for accounting purposes, but our system is unique because it allows restaurant management to manipulate and explore that data to inform their restaurant operations.

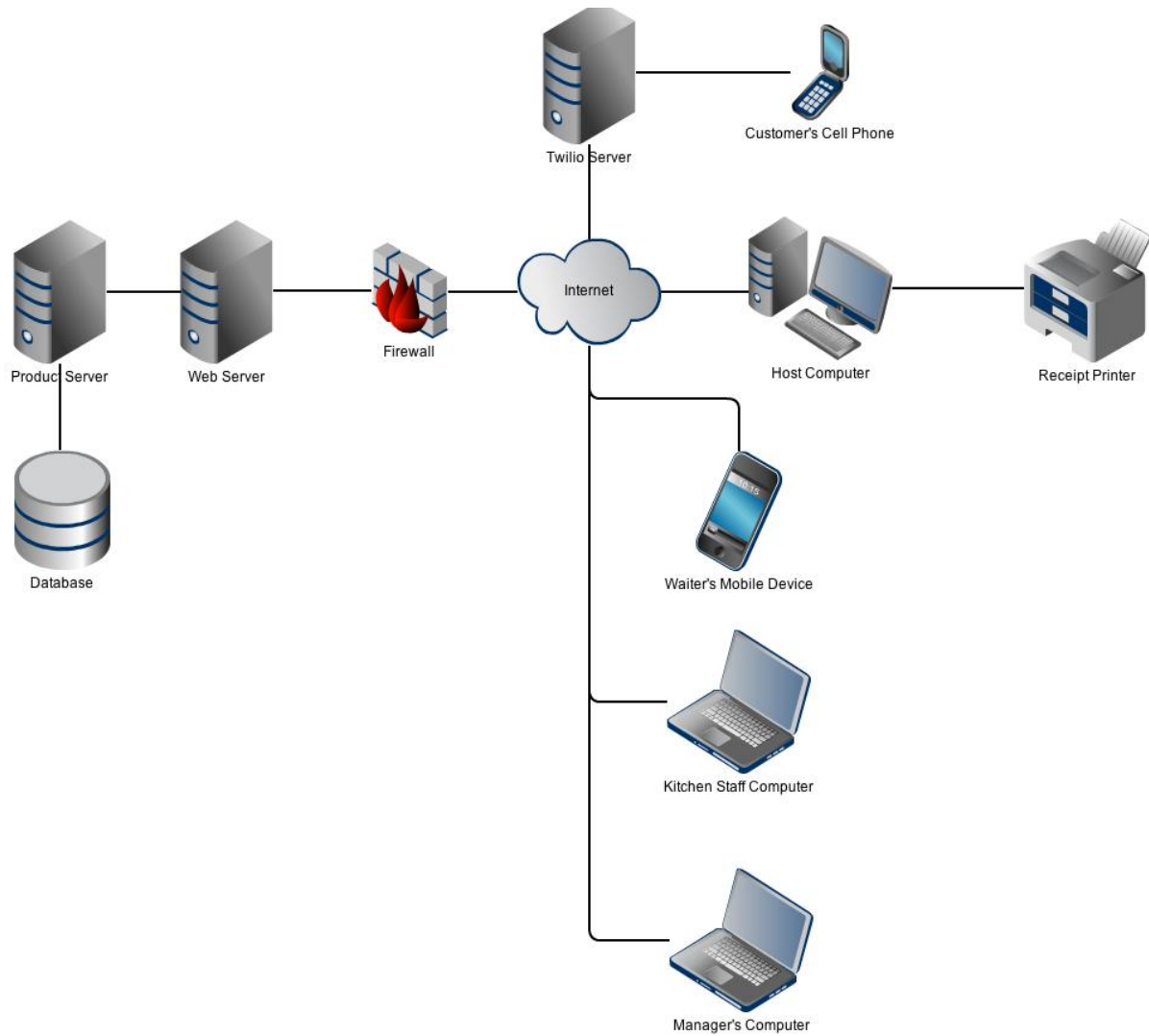
## SAMPLE WORKFLOW



# Part 2

## DESIGN (4)

### SYSTEM ARCHITECTURE DIAGRAM AND DESCRIPTION (4A)



## MINIMUM VIABLE PRODUCT (MVP) (4B)

The main goal of this product is to help restaurant personnel coordinate their activities and improve their services, and to allow managers to analyze business data. Given those goals, the minimum viable product for our system allows managers to handle the menu, staff, and review revenue over time. It allows hosts, wait staff, and kitchen staff to coordinate throughout customers' experience by ensuring they all have the same data about customers' tables and orders.

### A LEVEL USE CASES

#### All Staff

- All staff can log in and out
- Based on the person logging in, features will be separated and only management can access all of the features

#### Hosts

- Hosts can see the layout of the tables to see which are taken and which are not
- Host can click on a table to mark as occupied when customers are seated

#### Wait staff

- Wait staff can enter a customer's order with notes
- Wait staff can receive a notification on their device when customer's orders are prepared
- Wait staff can add items to customer orders or resend orders after wait staff has submitted an order

#### Kitchen Staff

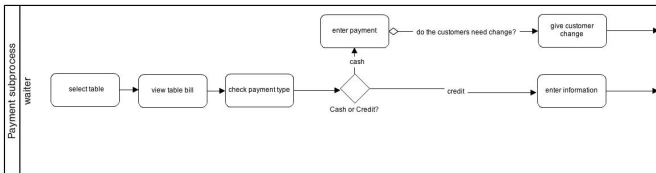
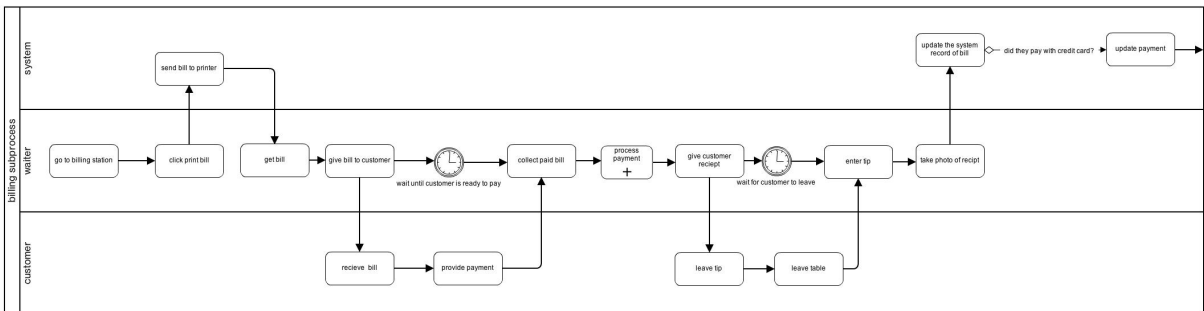
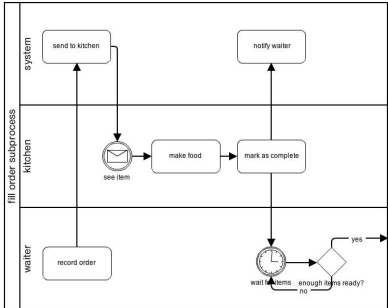
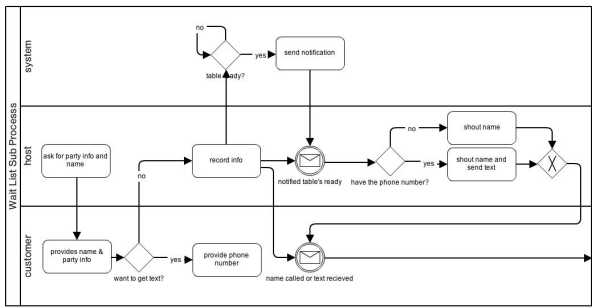
- See all pending orders in chronological order of when they were submitted
- Mark orders as complete

#### Management

- See all staff records and activities
- Edit the menu
- Add new staff members to the system
- See the revenue for a given time period



## SUBPROCESSES



STORYBOARD/WIREFRAMES OF SYSTEM IN CUSTOMER EXPERIENCE PROCESS (4D)

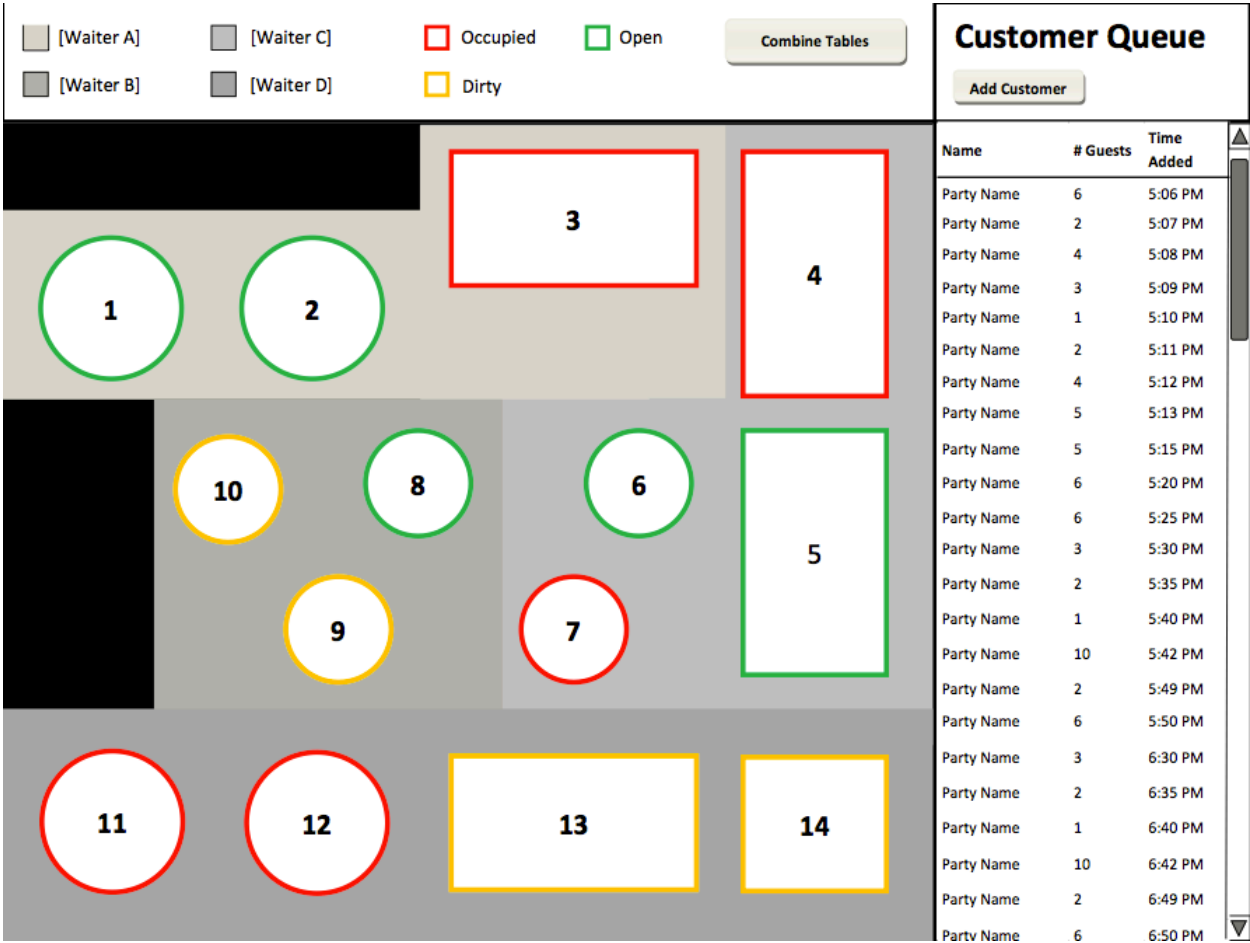


Figure 1 Hosts View of Tables and Waitlist



[Waiter A]

[Waiter B]

[Waiter C]

[Waiter D]

Occupied

Dirty

Open

Combine Tables

1

2

3

4

10

8

6

9

7

5

11

12

13

14

Customer Queue

New Customer

Name

# Guests

Phone # (optional)

Notes

Add Customer

Name	# Guests	Time Added
Party Name	6	5:06 PM
Party Name	2	5:07 PM
Party Name	4	5:08 PM
Party Name	3	5:09 PM
Party Name	1	5:10 PM
Party Name	2	5:11 PM
Party Name	4	5:12 PM
Party Name	5	5:13 PM
Party Name	5	5:15 PM
Party Name	6	5:20 PM
Party Name	6	5:25 PM
Party Name	3	5:30 PM
Party Name	2	5:35 PM
Party Name	1	5:40 PM
Party Name	10	5:42 PM
Party Name	2	5:45 PM

Figure 2 Adding a Customer to the Waitlist

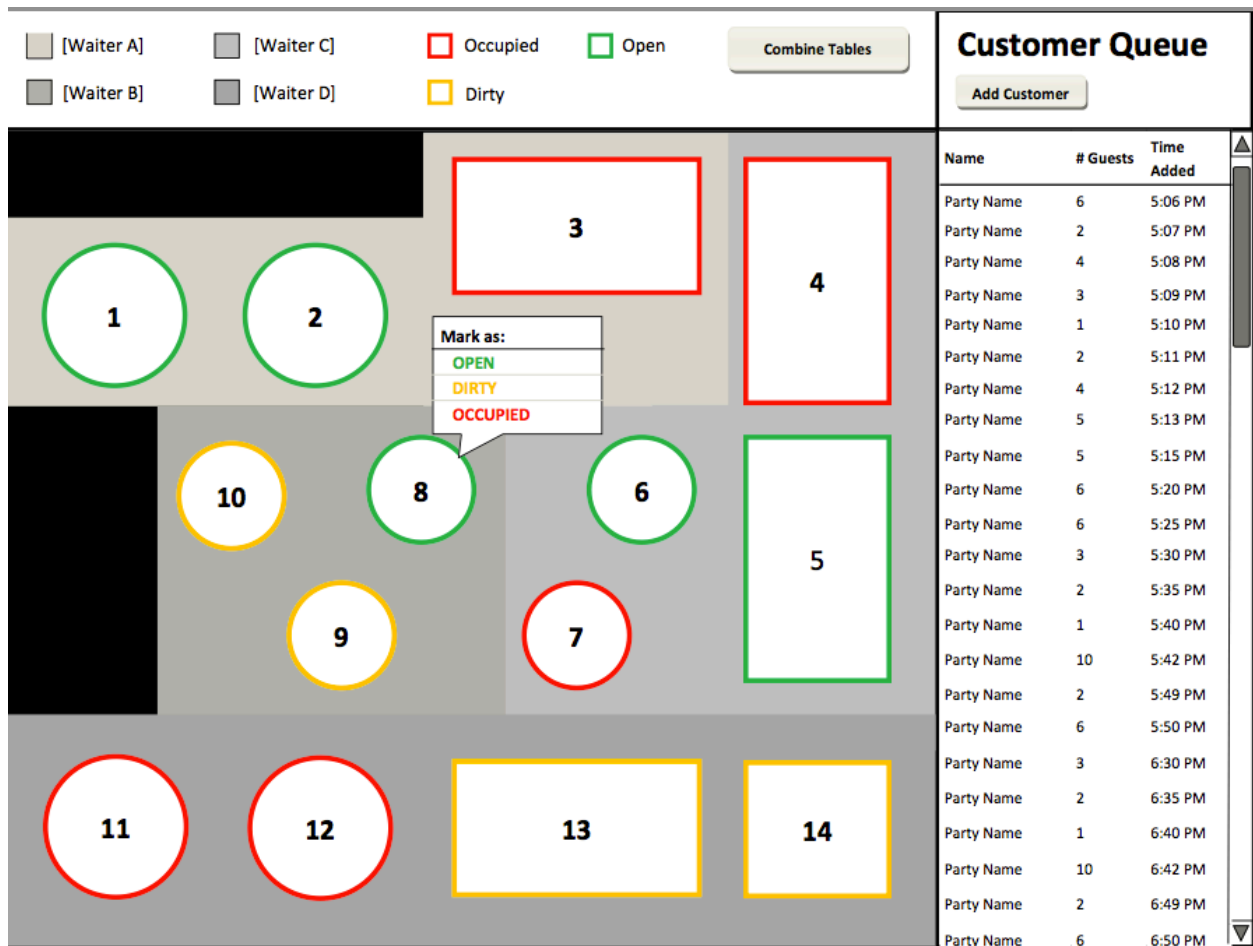


Figure 3 Updating Table Statuses

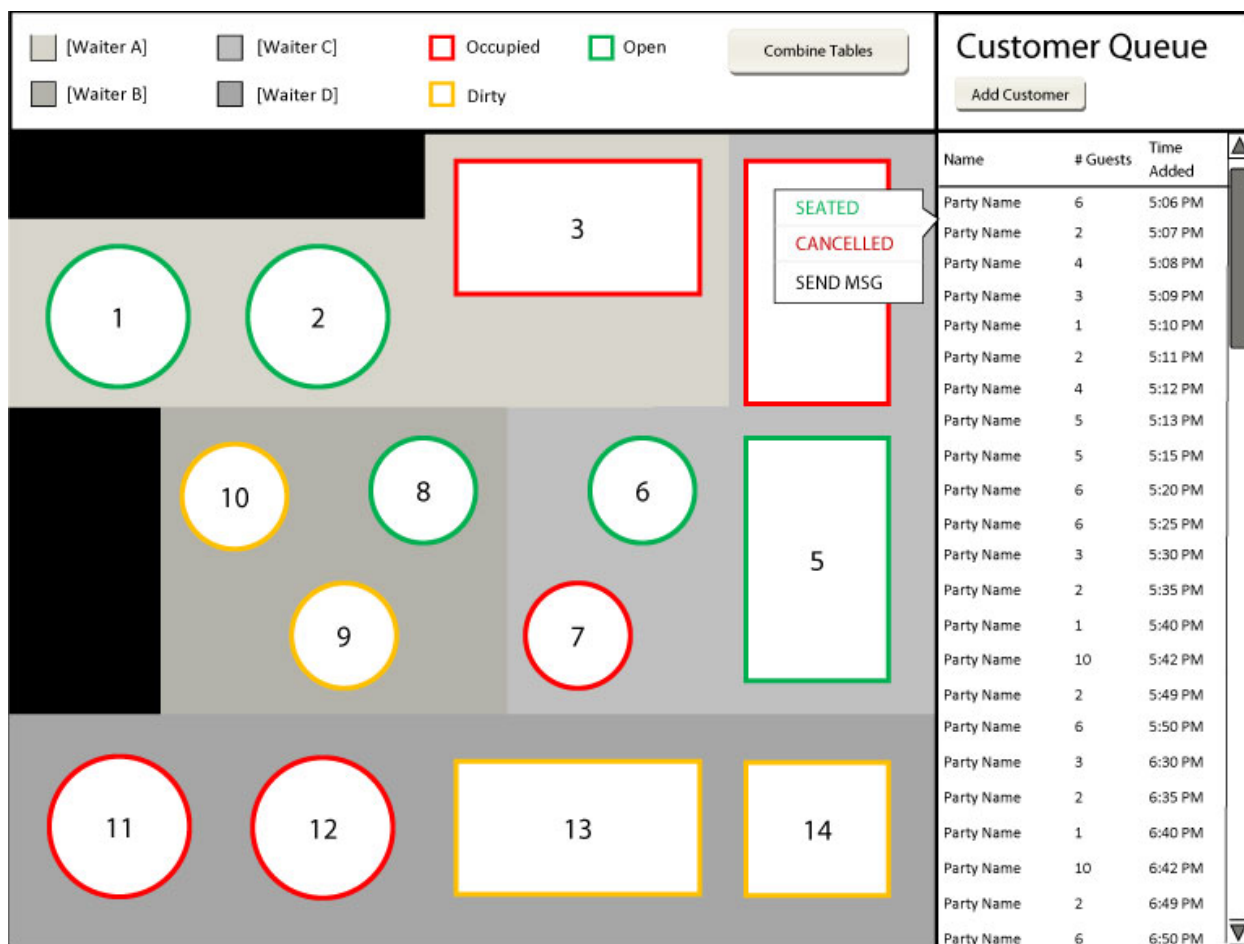


Figure 4 Manage Customer Parties on Waitlist

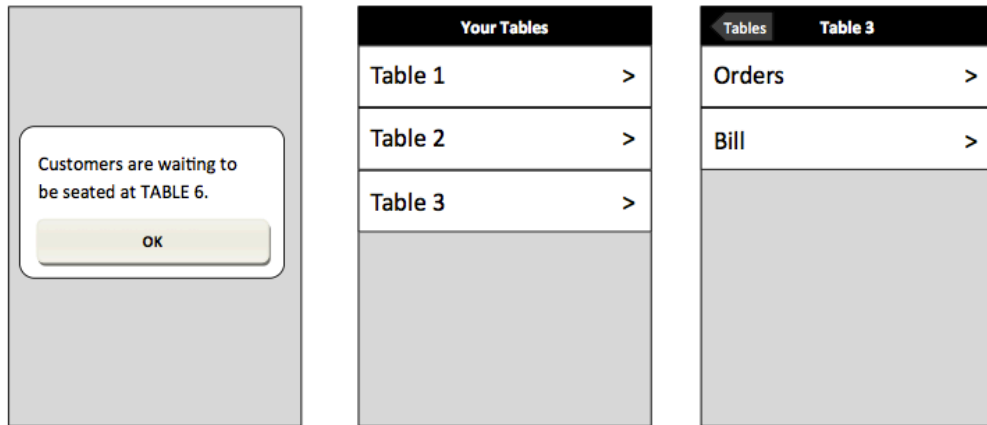
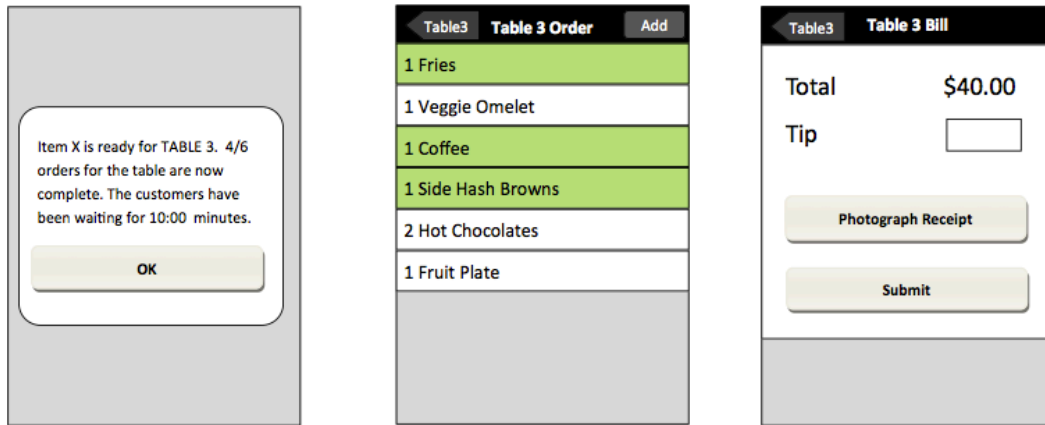


Figure 5 From left to right, waiter's notification to seat customers, table s list, and single table view.

Table 3Take Order	Your TablesDone	BackYour TablesDone	BackYour TablesDone
Orders >	Search	Search	Search
Bill >	Menu Section A >	Menu Sub-section A >	Menu Item X +
	Menu Section B >	Menu Sub-section B >	Menu Item Y +
	Menu Section C >	Menu Sub-section C >	Menu Item Z +

Figure 6 Adding menu items to an order.



**Figure 7** From left to right, an item ready notification, view order screen (with option to add to the order), and "add tip" page. All part of the waiter's interface.

Pending Bills

Table 1

Table 2

Table 3

Table 4

Table 5

Table 6

Bill for Table 3

1 Fries	\$5.00
1 Veggie Omelet	\$10.00
1 Coffee	\$2.00
1 Side Hash Browns	\$4.00
2 Hot Chocolates	\$5.00
1 Fruit Plate	\$4.00

SUBTOTAL	\$30.00
TAX	\$3.00
TOTAL (minus tip)	\$33.00

Print

Pay

Card >

Cash >

Figure 8 Viewing the Bill for a Table

Pending Bills

Table 1

Table 2

Table 3

Table 4

Table 5

Table 6

Bill for Table 3: Credit Card Payment

Swipe Card

...

Charge \$33.00 to Mastercard number \*\*\*\*\*1234?

Cancel

OK

Figure 9 Processing Credit Card Payments



Pending Bills

Table 1

Table 2

Table 3

Table 4

Table 5

Table 6

Bill for Table 3: Cash Payment

Cash Amount

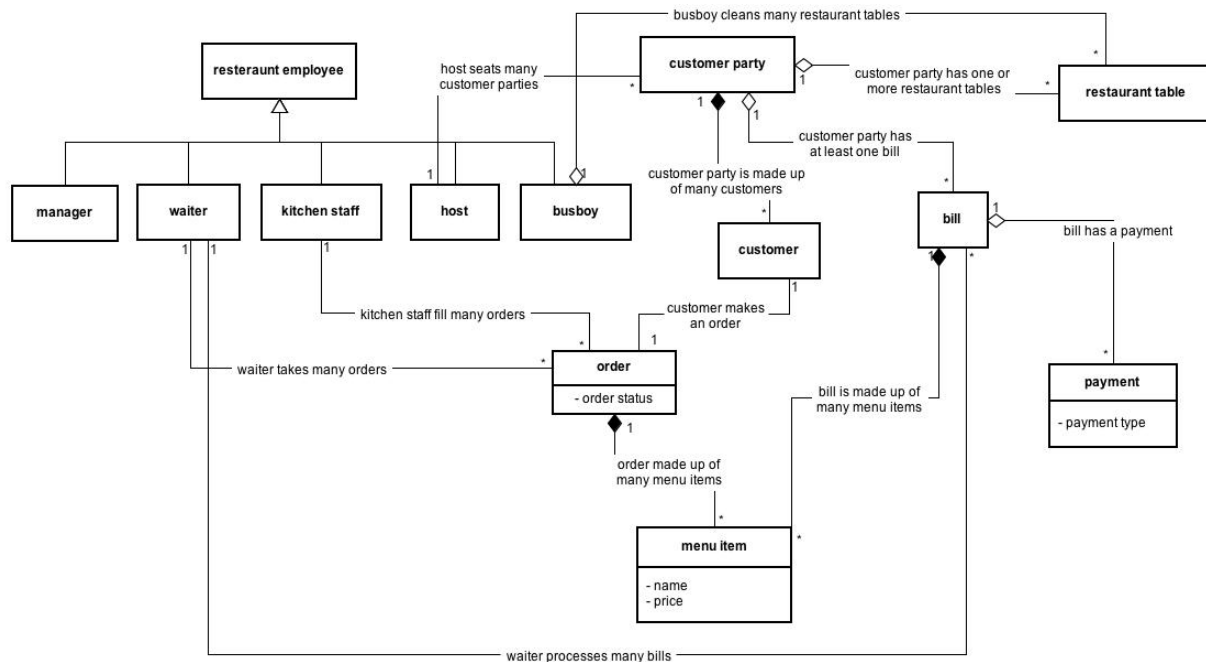
The bill is \$33.00. The customer paid \$40 in cash, and needs \$7.00 in change.

OK

Figure 10 Processing Cash Payment

## SYSTEM OBJECTS (4E)

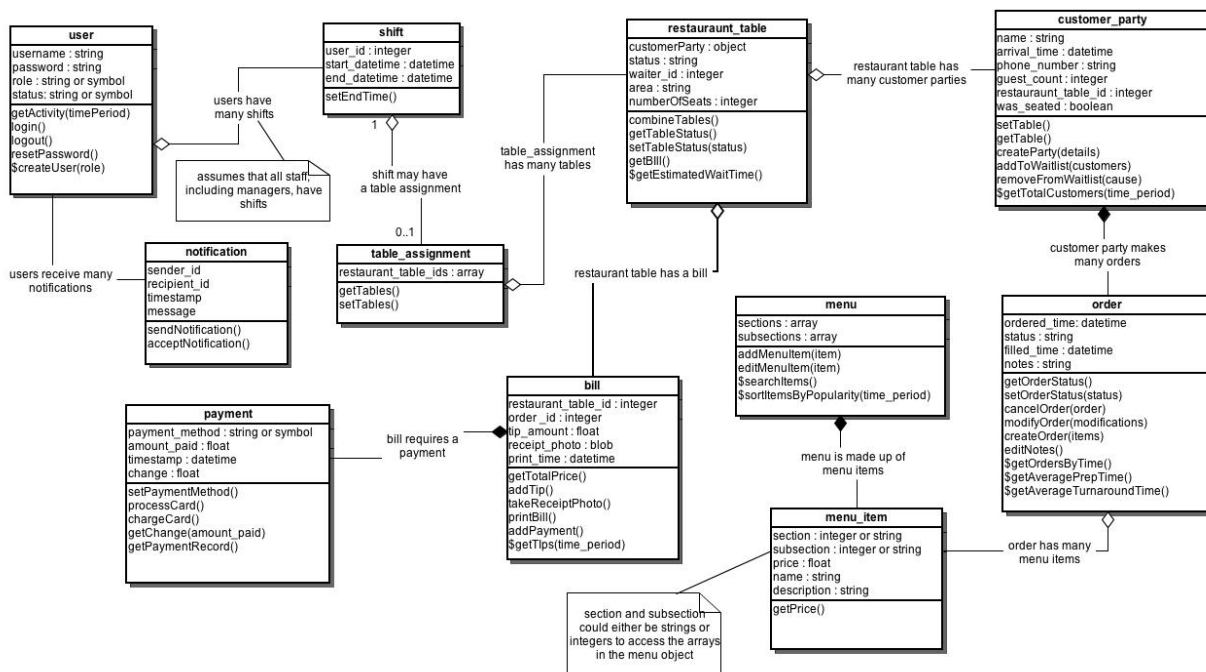
### CONCEPTUAL (DOMAIN) CLASS DIAGRAM (A)



create and share your own diagrams at [gliffy.com](https://gliffy.com)



### IMPLEMENTATION CLASS DIAGRAM (B)



create and share your own diagrams at [gliffy.com](https://gliffy.com)



## ROI ESTIMATE (5)

### COMMENTS/SUMMARY

The total cost to develop this system from scratch would be approximately \$129,800 based on our estimates. Including hardware (if needed to be bought) the cost of the system would be around \$136,167. Per client, yearly maintenance would only cost \$50, mainly towards hosting and data storage costs.

The current system loses money for the restaurant in a couple of ways. First, the system reduces customer loyalty losses currently caused by order fulfillment delays and overbilling. Also, money is lost through underbilling. We estimate that the current system causes \$98,112 in customer loyalty loss (due to time delays and billing errors) each year, and \$36,792 in losses due to underbilling. If our system reduced billing errors up to 85% and time delays by 60%, we estimate that there would be \$481,973 in benefits over the next three years. Since the total system costs \$136,316, our calculated ROI is 203%.

**See attached spreadsheet**

## EXECUTIVE SUMMARY (7)

### INTRODUCTION

This system is made to computerize the operations of small restaurants currently using paper-based systems. One example of a restaurant that would be in our target audience is the Pittsburgh restaurant Pamela's. The system helps to coordinate staff and manage of records. The system should help management lower the operating cost for the restaurant and use data to increase growth.

### REQUIREMENTS

The stakeholders of the system include the host, wait staff, kitchen staff, busboys, management, and customers. The host uses the system to view and update the status of tables and to manage the waitlist. Wait staff enter orders and get notifications when items are prepared from kitchen staff. Waiters also use the system to handle payments and tips. Kitchen staff tracks the order queue and check off items once they are ready. Busboys use the system track and update which tables are clean. Management uses this system to manage staff and the menu, and to review business data. Customers do not interact with the system, but because their payment information is entered in the system, they require it to be secure.

### COMPETITIVE ANALYSIS

We reviewed two existing mobile Point of Sale (POS) systems, AccuPos and Revel. AccuPos is Android-based while Revel is iPad-based. Our system is different from these others in several ways. First, it is web-based so that restaurants can choose what type of hardware they want to use; any device will work, as long as it has a web browser. Also, only our system supports automatic notifications to help employees coordinate their work and activities. Last, only our system provides a web-based dashboard for management to review key performance indicators.

### DESIGN

Our system has multiple touch points throughout the customer experience. It helps the host manage the waitlist and notify customers when a table is ready. Waiters record orders and deliver them to the kitchen on mobile devices (e.g., an iPod touch). The system automatically informs kitchen staff when an item is ready. The system also processes payments and allows waiters to add tips to a bill from a mobile device. Finally, bus boys use the system to track and update the status of dirty and clean tables, and the system automatically keeps the host's display up do date.

## ROI

We estimate that the current system causes \$98,112 in customer loyalty loss (due to time delays and billing errors) each year, and \$36,792 in losses due to underbilling. If our system reduced billing errors up to 85% and time delays by 60%, we estimate that there would be \$481,973 in benefits over the next three years. Since the total system costs \$136,316, our calculated ROI is 203%.

## TEAM STRUCTURE

Our team was comprised of Julia Teitelbaum, Rintaro Sato, and Rupa Patel. At the beginning of the project, we decided to not have formal roles on the team. Our rationale for this was:

- we did work together in work meetings, so that we didn't need someone to assemble components/make them consistent
- we wrote the project log together
- we were such a small team that it didn't make sense to have a project manager; instead we shared responsibility

## MEETING 1

Date: 11/13/12

Start Time: 1:30PM

End Time: 2:48PM

Attendees: Julia, Rupa, Rin

## AGENDA

- outline the report
- brainstorm for the system
- start on the requirements and due diligence so we can show it to TA

## MINUTES

- Making outline for report
- Brainstorm for system
- discussed the stakeholders
- requirements
- quality attributes scenarios
- quality attribute requirements
- competitors

\* This was a work meeting, so these areas we worked on, and rather than writing minutes, we added content to our report.

## ACTION ITEMS

- Plan next meeting date and time
- Plan meeting with TA (Lizzie)
- Proofread google doc of report

## MEETING 2

Date: 11/15/2012

Start Time: 12:40

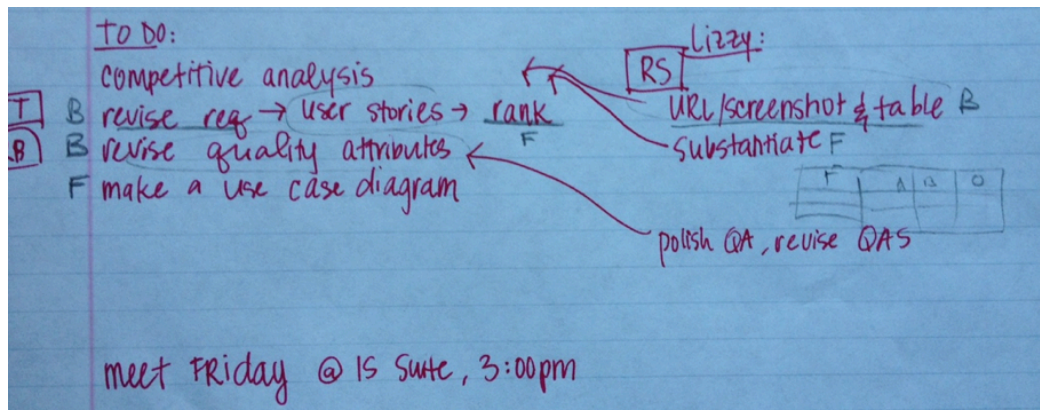
End Time: 1:20

Attendees: Julia, Rupa, Rin

## AGENDA

- plan upcoming work and meetings
- decide what work to do offline and delegate it

## MINUTES/ACTION ITEMS



## MEETING 3

Date: 11/16/2012

Start Time: 3:00

End Time: 5:30

Attendees: Julia, Rupa, Rin

## AGENDA

- discuss what we worked on last night
- write up description of competitors
- draw up use case diagram using Gliffy

## ACTION ITEMS

- Julia
  - add mgmt to use case diagram
  - edit 1b after rupa writes
  - share email to lizzie with questions and email
- Rin
  - share use case diagram
  - finish typing up whiteboard

- add to these meeting minutes
- Rupa
  - draft 1b
- For later
  - double check use case diagram against guidelines in Raja's email "[67371-f12] phase-1 UCM"

## MEETING 4

Date: 11/20/2012

Start Time: 1:12

End Time: 2:30

Attendees: Julia, Rupa, Rin

## AGENDA

- Discuss TA comments
- Put final touches on the report
- revise use case diagram

## MINUTES

- Fixed use case diagram
- Checked use case with requirements
- Added customers as primary stakeholder
- Revised the overall report

## ACTION ITEMS

- Julia
  - move report to word document for formatting
  - fix wording in places
- Rin
  - make use case diagram more readable
  - figure out what we're doing for the next phase
- Julia
  - outline next part of report
- Rupa
  - start looking into cost/estimating preliminarily (5)
- Rin
  - add notes about things like mvp and system components to outline (4a)
    - remember the insulin pump and garage door exercises
  - schedule next meeting



## MEETING 5

Date: 11/28/2012

Start Time: 6:30

End Time: 7:15

Attendees: Julia, Rupa, Rin

### AGENDA

- Discuss what to do as TA
- discuss MVP
- discuss architecture diagram
- discuss system object

### MINUTES

- MVP
- set up meeting time for Friday at 1:00pm
- next meeting- whiteboard bpmn, system architecture diagram

### ACTION ITEMS

- Julia
  - description of MVP
- Rupa
  - implementation diagram
- Rin
  - ROI
  - figure out what we're doing for the meeting on Friday

## MEETING 6

Date: 11/30/12

Start Time: 1:00 PM

End Time: 2:30 PM

Attendees: Julia, Rupa, Rin

### AGENDA

- Sketch system architecture diagram
- Sketch business process diagram

### MINUTES

- Gliffy diagram

## ACTION ITEMS

- Julia & Rupa
  - meet again later to sketch storyboards
- Rin
  - add arrows to gliffy business process diagram
- Julia & Rin
  - figure out next steps for ROI
- Rupa
  - upload photos of business process diagram sketch

## MEETING 7

### AGENDA

- Sketch storyboards

### MINUTES

- [Photos of Sketches](#)

## ACTION ITEMS

- Rin
  - ROI by Monday midnight
  - finish system architecture diagram
- Rupa
  - site map
  - if problems, talk with Julia
  - if you can, start domain modeling
- Julia
  - Clean up business process diagram
  - computerize/finish storyboards
- Next meeting
  - monday, 4:30
  - domain modeling
  - just julia and rupa
    - after class on Tuesday

## MEETING 8

Date: 12/02/2012

Start time: 4:30

End time: 6:10

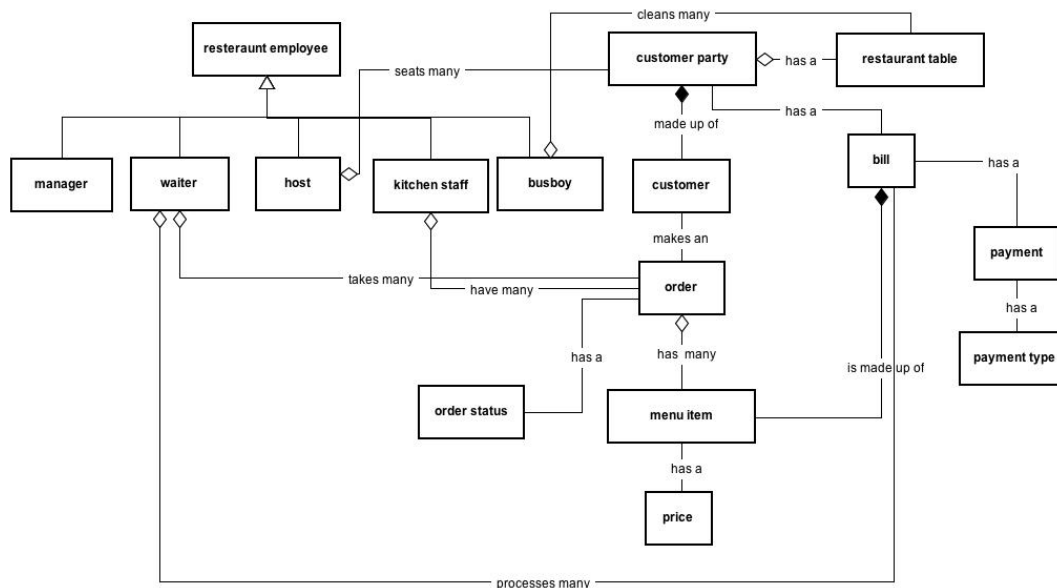
Attendees: Julia, Rupa

## AGENDA

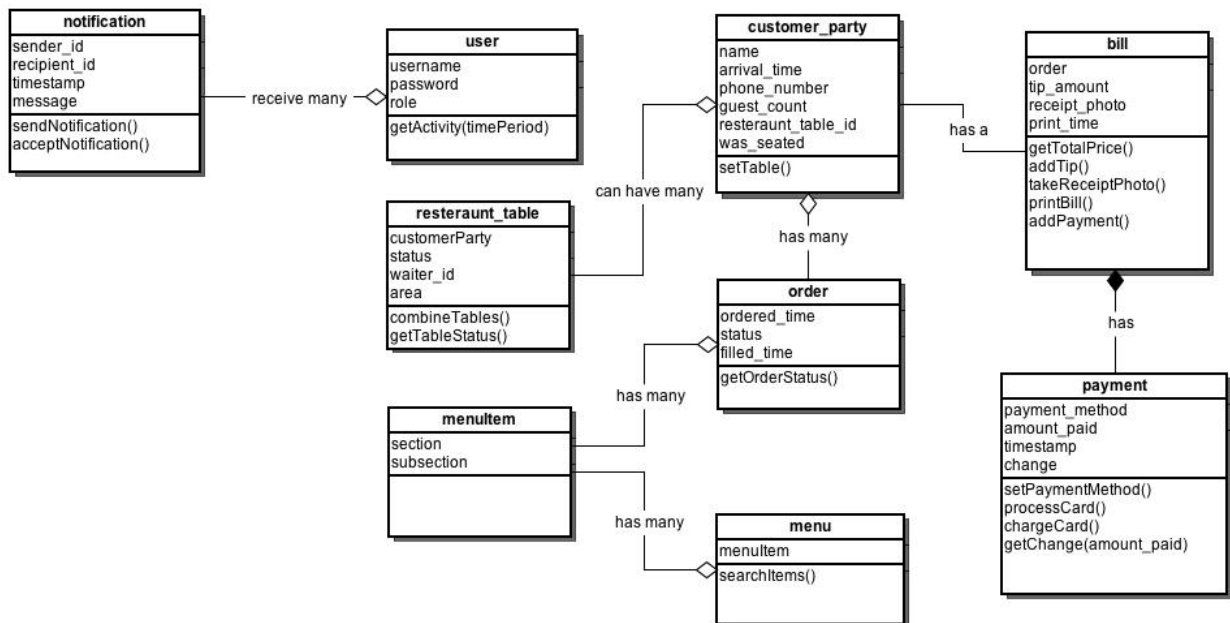
- Work on domain model
- Work on implementation model

## MINUTES

### DOMAIN MODEL



### IMPLEMENTATION MODEL



## ACTION ITEMS

- wait on Lizzie's feedback
  - revise things tomorrow
- domain model
- implementation model
- mvp
- justification?
- ROI
  - ROI calculations as well as cost
  - move to excel
- business process diagram
- possibly adding to storyboards
- share system architecture stuff - **Rin**
- write executive summary

## MEETING 9

Date: 12/04/2012

Start time: 12:30

End time: 2:30

Attendees: Julia, Rupa, Rin

## AGENDA

- look through lizzies email to revise domain model, bpmn
- overall revise

## MINUTES

- revise domain model
- revise Implementation Model
  - send to lizzie
  - used requirement for whole system not just mvp
  - we removed 2 use cases from part 1
- revise BPMN
- revise MVP
- revise system architecture

## ACTION ITEMS

- continue revising models

## MEETING 10

Date: 12/04/2012

Start Time: 2:30

End Time:

Attendees: Rin, Julia

## AGENDA

- Polish ROI spreadsheets

## MEETING 11

Date: 12/05/2012

Start time: 4:30

End time: 9:00

Attendees: Julia, Rupa

Rin- Late came at 5:00

## AGENDA

- finish

## MINUTES

- finished BPMN
- finish ROI
- finish executive summary
- finish report

## ACTION ITEMS

- eat and drink
- be merry