Auto Order 2.0

Group 2

Report #1: System Specification

Website: https://sites.google.com/site/restautomation/

"The hardest single part of building a software system is deciding what to build. No part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later."

– Fred Brooks

Group Members:
Praveen Chekuri

Pradnya Pisal

Zachary Brown

Kartik Bhatnagar

Bill Fung

Individual Contribution Breakdown

Task/Group Member	Praveen	Pradnya	Bill	Kartik	Zac
Project Management (10 points)	5		5		
Sec 3: Customer Statement of Requirements (6 points)		**4**	2		
Sec 4: Glossary of Terms (4 points)			**4**		
Sec 5: Functional Requirements specification (37 points)	**7**	**7**	**9**	5	**9**
Sec 6: Nonfunctional Requirements (6 points)				**6**	
Sec 7: Domain Analysis (25 points)		9		**9**	7
Sec 8: User Interface Design (8 points)	4				**4**
Sec 9: Plan of Work (3 points)	**3**				
Sec 10: References (1 point)	1				

Individual Point Allocation

Praveen Chekuri	20
Pradnya Pisal	20
Bill Fung	20
Kartik Bhatnagar	20
Zachary Brown	20

Table of Contents

Customer Statement of Requirements

In order to effectively run a restaurant, cost and time saving are essential. Minimizing time by a few seconds for each table can speed up order processing, increase efficiency and boost profits. The biggest hurdle most restaurants face is the migration from a paper-pencil system to a completely automated touch-screen system. Most retail establishments choose to use a point-of-sale system (herein referred to as POS) to handle their transactions. For a restaurant, a POS system can greatly change the following problems plagued by typical 'pen & paper' establishments:[1]

- Keeping track of empty, clean and reserved tables within a restaurant. This requires a notebook or whiteboard which is constantly changing.
- Waiters making mistakes with customer's orders. At times, a waiter can forget to add a specific item, make a change due to a customer's allergies, or forget to give the order to the kitchen.
- Busboys need to keep track of which tables need clearing. This means that they must be always checking for tables. Waiters need to usually alert them. This takes extra time from other staff.
- Waiters need to constantly check with cooks to determine when food is ready. Conversely, cooks need to make sure waiters know that food is ready. This can create cold food (potential food-poisoning), wrong orders and an unsatisfied customer.
- Managers need to maintain record-keeping of employee hours for payroll. They must also re-print menus when food is not available or a price needs to be changed. This can be costly and time-consuming to a restaurant.
- Managers need to analyze hundreds of paper receipts to determine best-selling items, popular hours and customer satisfaction.
- Customers, when seated, have to wait for a waiter to order. They must rely on the waiter to remember their order and specific details. Their food may take longer to be prepared if the waiter has multiple tables. They may be billed incorrectly since they cannot see their check until their meal is complete.
- Typical reservation systems rely on host/hostess to mark down reservation requests using a notepad and marking the table during the time. This could lead to reservation discrepancies.
- Impatient customers also call over the waitress to find out the status of their order several times during their visit, wasting the waiter's service time.

To remedy these issues, a POS software solution is the best option. The POS system should have these basic high-level features[2]:

- Allow the restaurant to operate faster (faster turnaround on food, faster seating, faster order preparation)
- Reduce employee error (thereby increasing customer happiness). This also reduces waste (when the wrong item is ordered, food must be discarded).
- Operate on two hardware platforms:
  - Desktop touch-screen computers for employees. Employees will use the touch-screen computers to interface directly with the POS system.
  - Android tablets running a version of the POS system. The tablets will be provided to customers, at their tables, allowing them to directly order from the computer system installed in the restaurant. The tablets are the property of the establishment and are kept at each table.

---

[1] These are a variety of typical problems which the software should tackle. The software solution should at a minimal tackle these problems.

[2] A more detailed remedy is further prescribed in the CSR.

The architecture of the system shall appear as follows (connected lines illustrate a wired network connection, while curved linear near a device represent a wireless network connection):

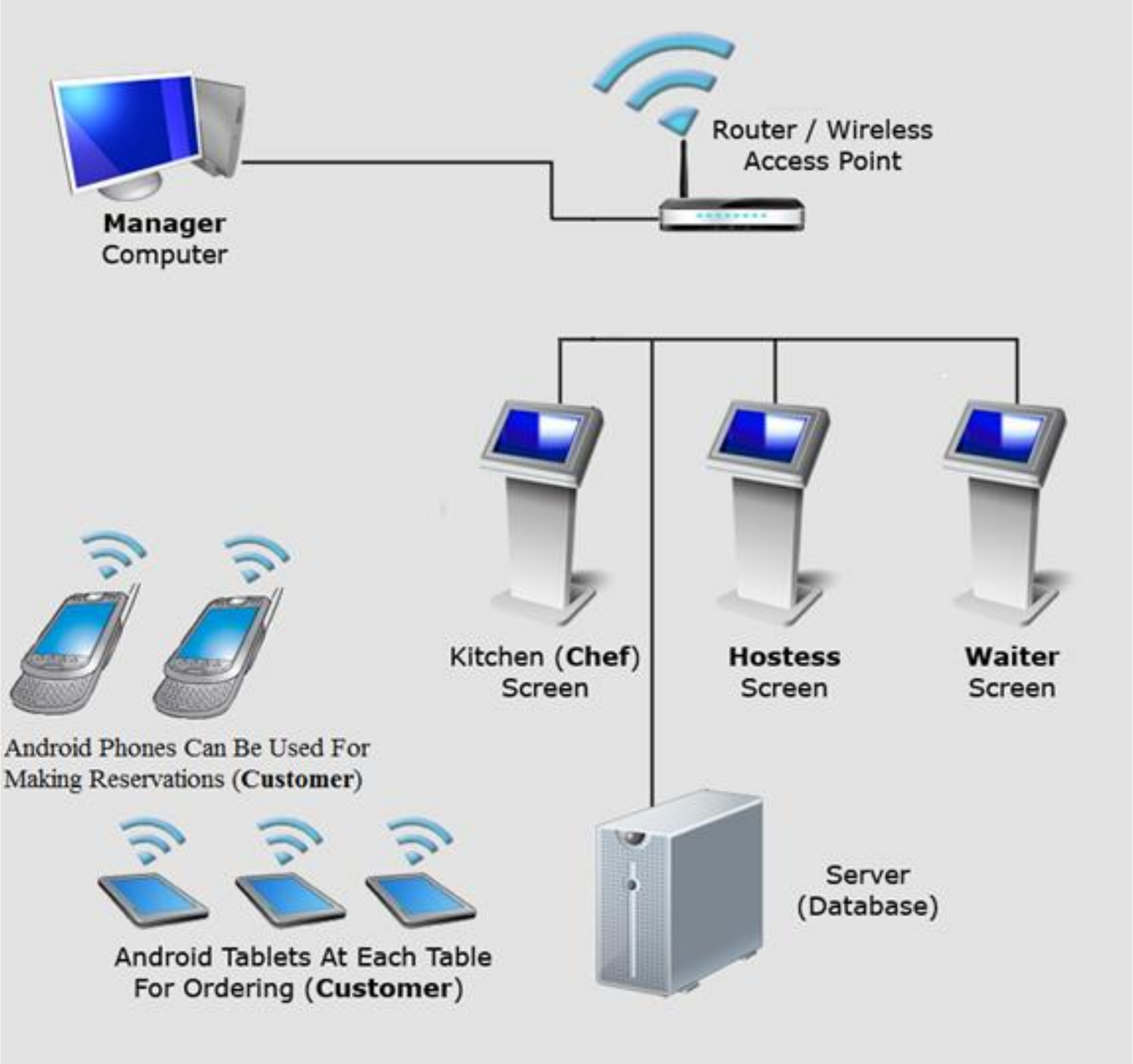

Router / Wireless Access Point
Manager Computer
Kitchen (Chef) Screen
Hostess Screen
Waiter Screen
Android Phones Can Be Used For Making Reservations (Customer)
Server (Database)
Android Tablets At Each Table For Ordering (Customer)

Fig 1 - CSR System Schematic

Detailed Requirements:

The system (architecture seen above), should provide the following specific functional requirements:

- Provide six users and abilities for each user role: Manager (Administrator), Chef, Hostess, Waiter, Customer.

Each of the user roles should meet the following functional requirements:

Manager**:**

- Create users which fill in each role. Manage users in each role (delete them, modify their pay, add comments, etc.)
- Manage users (view the time sheet of each employee and determine based on their hours their pay rate)
- Modify the menu of the restaurant. Add items, remove items, change their description, and update prices.
- Run reports on the restaurant (these can be static or graphical as necessary):
    - Determine which items are the most popular
    - Determine the most popular time of day for sales
    - Determine how long it takes for food to be prepared.
- Leave notes on employees (payroll, behavioral, etc. for the employee record)
- Modify the table layout of the restaurant (floor plan). This change will be reflected on the screens of all other restaurant users who have layout access abilities.
- Manage any administrative computerized aspects of the restaurant.
- Edit the advertisements displayed by the tablets during inactive periods.
- Has the authority to control customer's bills in order to provide discounts due to inconvenience.
- Login interface to prevent un-authorized access of restaurant management.

Hostess:

- View a layout of the restaurant
- Using a color coded system, red (occupied table OR being cleaned), yellow (reserved), green (open), be able to quickly identify tables where guests may be seated at
- Change the status of a table to "occupied (red)" when a customer sits at a table.
- Ideally, show the customer the restaurant layout, thereby allowing the customer to sit at his/her chosen table
- Ability to use 'timecard' functions for payroll purposes (e.g. clock-in and clock-out of shifts)
- Assign waiters to tables using their judgment based on waiter's experience and party size.

Chef:

- Queue of upcoming orders, organized by order most-recently submitted.
- Ability to determine the table number of the customer who ordered, as well as the waiter who is assigned to the table.
- When an order is prepared, a feature to "Complete" the order. The appropriate waiter will then be notified at their terminal.
- When an order is completed, the order will be removed from the queue. If there are additional orders (not on screen), the order will be fetched from the system and added to the screen.
- Chef's screen will have a login in order for manager to track chef's orders in case customers complain about poorly cooked food. There will be several terminals which will depend on the number of chefs working in the kitchen and feasibility of installing additional terminals in the kitchen.
- Interface should include a 'timecard' function for payroll

Waiter:

- Ability to access the tab of a table they are assigned to. Within the tab the waiter should be able to:

    - View the cost of specific items
    - Modify the tab of the table (e.g. provide complementary food in case the customer is not satisfied), add specific extra charges, or reduce the bill
    - Close the tab of a customer. When a customer is ready to pay, the waiter should be able to charge payment
        - ✓ Provide support for entering a credit card number (stored in database for processing later)
        - ✓ Open a cash drawer for cash
- Alerts when a table is seated, when a table 'calls them over' , and when a tab is requested to be closed.
- Notification system when food for one of their tables is ready.
- Quick login system. There may be multiple waiters sharing the same terminal. A drop down box of waiters, as well as a pin number for secure login (4 digits).
- Ability to use 'timecard' features, similar to other employees (payroll purposes).
- **In addition**, busboy duties will be replaced by waiters in order to save money for restaurant operation. The additional duties shouldn't be too demanding for the waiters as their duties will be reduced by the automation.
- Alerted when a table tab has been closed and a table needs to be cleaned.
- When a table is cleaned, change the status of a table. This in turn shall show on the floor map that a table is ready for use by the next customer.

Customer:

- Simple, easy to use, Android-based application on restaurant provided touch screen tablet. Use the table to place order in the restaurant computing system.
- View the menu of the restaurant and add menu items to cart
    - As items are added to 'cart', see total price, add notes (e.g. "Steak rare", or "no sauce")
    - Apply coupons or promotions to purchases.
    - Order 'cart' of food when ready. This order is sent to the chef.
- Call over waiter for question / comment on food.
- Request to close the bill for purchase and pay.
- Using their android device, customer should be able to reserve a table for a party of upto twenty people within a three-day time frame, atleast 24hrs prior to the period of reservation.

Due to the fast-paced nature of restaurant environments, the number of keystrokes, clicks, and touches on computer screens should be kept to a minimum. This will ensure an easy to used, efficient system. It is key to remember, the goal of the project is to create a system which is to replace an inefficient, antiquated paper and pen system. If the new system is less efficient, it has failed its design purpose. Functionality, security of the system must not be overlooked.

Glossary of Terms

- **Manager** – Controls and oversees all of the business. In charge of editing the menu items available to the customers. The manager is also responsible for assessing restaurant performance.
- **Customer** – Orders Food and services from the restaurant.
- **Chef** – Cooks food ordered by the customer.
- **Waiter** – Assists customer upon request. Waiter is responsible for serving food.
- **Hostess -** Receives customers and handles reservations.
- **Clock In/Clock Out** – A system function that allows employees to enter their hours for manager to run payrolls.
- **Reservation System** – Customers can reserve a table using an android application.
- **Payroll** – The amount of money each employee will receive for their services. The payroll differs for each employee.
- **Menu** – List of dishes available in the restaurant. Will be displayed on an android tablet.
- **Floor Layout** – Shows all tables in the restaurant along with their respective status.(ex: Occupied, Reserved etc.)
- **User Interface** – The visual on the computer and tablet that allows user interaction with the system. Allows touch screen interaction to order, pay bill, call waiter and view order.
- **Table Status** – Shows the availability of a table as well as the waiter in charge of serving that table.
- **Order Status** – Shows whether the order of a particular table is ready to be “served” or “cooking”.
- **Payment** – Customers have the option of payment through cash or credit.
- **Ads** – Appear on the menu when idle. Manager has options to edit the slides.
- **Wireless Access Point** – Access points used in home or small business networks are generally small, dedicated hardware devices featuring a built-in network adapter, antenna, and radio transmitter. Access points support Wi-Fi wireless communication standards.
- **Terminals** – These are terminals used by hostess, manager, chef and waiter running Windows 7.
- **POS** – Point of Sale System

System Requirements

Functional Requirements

Requirements	PW	Description
REQ1	5	The system should allow employees to login given a unique identification number to access their respective interfaces.
REQ2	5	The system should be able to modify the employee personnel list.
REQ3	5	The system should allow orders requested from the android tablet menu to be sent to chef for processing.
REQ4	2	The system should handle a customer's request to reserve a table.
REQ5	3	The system should display a real time floor plan of the entire restaurant by allowing the hostess and waiters to update the table status.
REQ6	5	The system should allow a customer to notify a waiter for assistance through the android tablet.
REQ7	5	The system should store the working hours of all employees.
REQ8	1	The system should allow the chef to update the status of each order.
REQ9	2	The system should store all menu necessities.
REQ10	4	The system should keep track of all income from restaurant activities.

Non-Functional Requirements

Requirements	PW	Description
REQ11	4	The cook's user interface should notify the waiter on completion of an order with 1 screen touch of the root window.
REQ12	2	The host's screen should prompt for the reservation confirmation code when a reserved table is selected.
REQ13	5	Managerial tasks should be limited to the manager's user interface to prevent potential security breaches.
REQ14	2	The customer should be able to make a reservation within 6 screen touches through their android interface.
REQ15	2	The login interface should be consistent for all restaurant personnel to maintain consistency in case an employee changes role.
REQ16	4	The cook's user interface should have reliability so not more than one order fails on a given day due to system failure.
REQ17	5	The menu (Android Interface) should contain text and graphics that clearly describe each item to an average customer.
REQ18	3	The host's interface should efficiently predict waiters for incoming customers so each waiter has the same workload.
REQ19	2	The reservation system should be compatible with any android phone provided it is connected to the internet.
REQ20	4	The waiter's interface should be able to distinguish between alerts from chef, customer and host clearly from a distance of 1 meter provided he/she has average eyesight.

Appearance Requirements

Requirements	**PW**	**Description**
REQ21	5	The host’s screen should display the floor layout replicating the actual layout of the restaurant.
REQ22	4	The menu should have a brief description of the item when selected.
REQ23	5	The chef’s screen should display two-three orders simultaneously so the chef knows what to expect once the current order is closed.
REQ24	3	The waiter’s screen should have the tables he/she is serving along with order details and chef updates.
REQ25	2	The tablets (menus) should run a slideshow of attractive ads.

Functional Requirements

Stakeholders

There are six key stakeholders involved in this system: customers, end users, owners of the system, project managers, system analysts, system architects and developers. In our system we, Group 2, will be the system analysts, project managers, system analysts, system architects and developers since we will be performing all the tasks required starting from system design until it is completely implemented and functional. The rest of the stakeholders have different stakes according to their role.

The end users will be those interested in the system's functionality. In our system, the manager, host, waiter and chef are all involved in the daily functions performed by the system. The customers also have comparable stakes in the system as they will be interacting with the system to the same degree as the end-users. Customers also have The owners of the restaurant are important stakeholders in the system as well since they will be investing in our system and any sort of failure of the product will cause them a loss of revenue.

Actors & Goals

Actors	**Goal**	**Use Cases**
Manager	To oversee all restaurant operations. Responsible for all employee payroll performance records. In charge of updating the menu through the addition and removal of menu items.	UC-1, UC-2, UC-3, UC-4, UC-5, UC-6, UC-7, UC-8, UC-9, UC-10, UC-11, UC-12, UC-13, UC-14
Customer	To order and pay for food and services.	UC-16, UC-19, UC-20, UC-21, UC-24
Chef	To prepare and update the status of food ordered by the customer	UC-1, UC-2, UC-18
Waiter	To assist the customer upon request. Responsible for serving food, cleaning tables, and updating the status of a table from unavailable to available	UC-1, UC-2, UC-17,UC-21, UC-22
Hostess	To oversee the activity of the restaurant. Responsible for seating customers, assigning waiters to designated areas and updating the status of a table.	UC-1, UC-2, UC-9, UC-18

Use Cases

1. Login (UC-1) Staff enters unique ID and password information which is verified by the system.
Fulfills REQS: 1

2. EnterHours (UC-2) Employees record the number of hours they have worked each day.
Fulfills REQS: 7

3. ManageEmployees (UC-3) Manager adds, remove, or modifies each employee in the database.
Fulfills REQS: 2

4. AddEmployee (UC-4) Manager adds a new employee in the database.

5. RemoveEmployee(UC-5) Manager removes an existing employee from the database

6. EditEmployee(UC-6) Manager edits employee information.

7. ManageMenu (UC-7) Manager adds, remove, or modifies items on the menu.
Fulfills REQS: 9

8. AddMenuItem (UC-8) Manager adds a new item in the menu.

9. RemoveMenuItem(UC-9) Manager removes an existing item from the menu.

10. EditMenuItem(UC-10) Manager edits menu item information.

11. ManageLayout (UC-11) Manager adds/removes tables and edits the floor plan of the restaurant. Fulfills REQ 5

12. ManageADs (UC-12) Manager adds, remove, or modifies ads of the slide show while the menu is idle. Fulfills REQS: 9

13. ViewPayroll (UC-13) Manager views hours logged by each employee and the wages owed based on these hours. Fulfills REQS:7

14. ViewSales (UC-14) Manager views sales records by month, week, and day.
Fulfills REQS:10

15. TableStatus (UC-15) Hostess assigns waiters to tables based on availability.

16. OrderFood (UC-16) Customer views menu and orders desired meals.

17. CancelOrder(UC-17) Waiter cancels an item in Order already sent to chef.

18. ServeFood (UC-18) Chef notifies waiter when food is ready to be served.

19. CallWaiter(UC-19) Customer calls waiter for assistance.

20. PayBill (UC-20) Customer views total cost of ordered meals and calls waiter to proceed to bill payment.

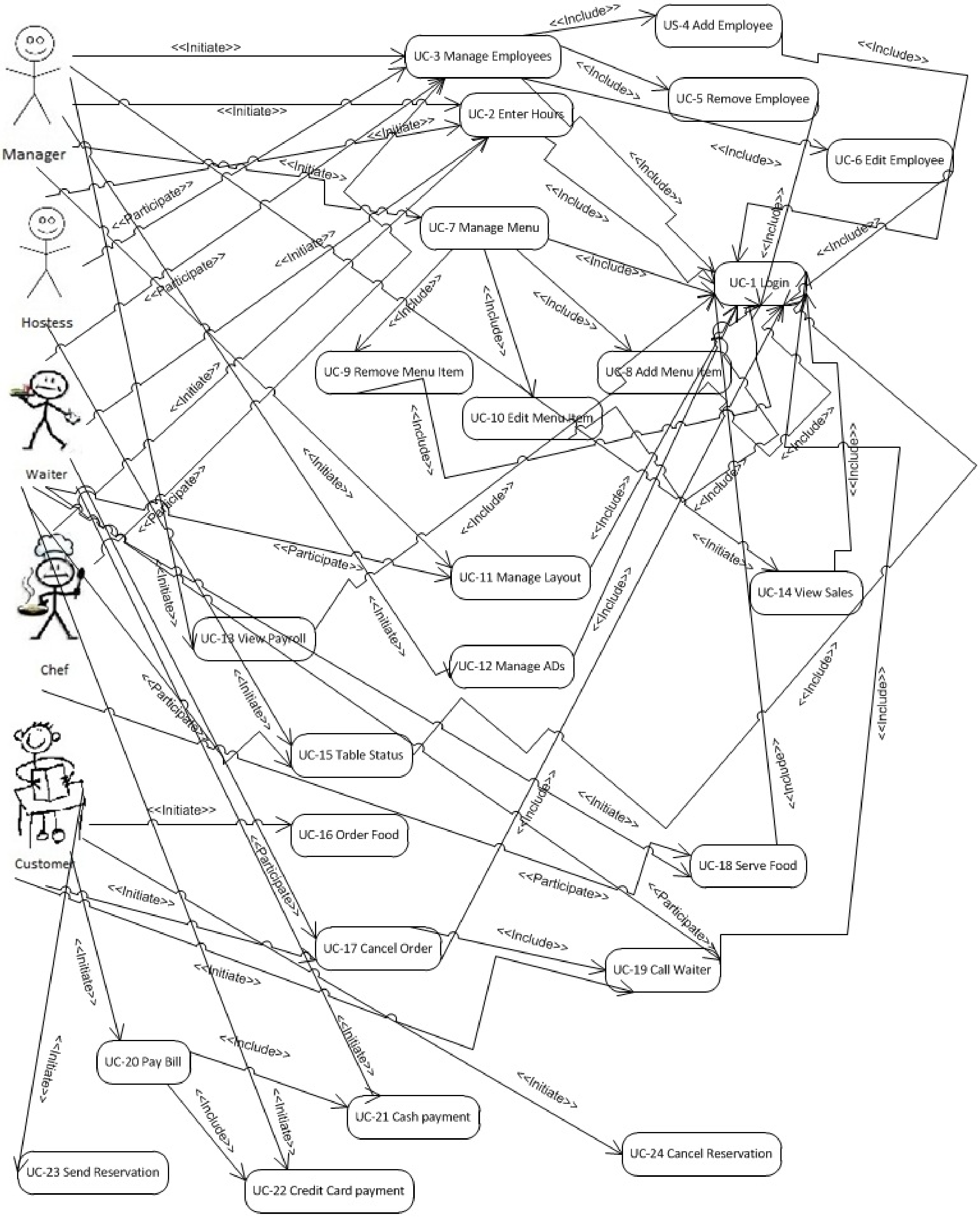
21. CashPayment (UC-21) Waiter pays for ordered meals using cash payment.

22. CardPayment (UC-22) Waiter pays for ordered meals through card payment.

23. SendReservation (UC-23) Customer views available reservation times and then sends a request for a reservation.

24. CancelReservation (UC-24) Customer cancels reservation.

Use Case Diagram



Fully-Dressed Descriptions

Use Case UC-1	**Login**
Related Requirements:	REQ1
Initiating Actor:	Hostess, Waiter, Manager, Chef
Actor's Goal:	To successfully login to the system and perform restaurant operations
Participating Actors:	
Preconditions:	Access to a computer connected to system network
Postconditions:	User successfully logged in to system
Failed End Condition:	Login Failed

Flow of Events for Main Success Scenario:

→ 1. User selects Login button
← 2. System prompts for a unique identification number
← 3. User enters information into designated field
← 4. System validates user input and displays proper interface of that user

Flow of Events for Extensions (Alternate Scenarios):

4 a. System could not validate log information and sends a login failed error

Use Case UC-2	**EnterHours**
Related Requirements:	REQ7
Initiating Actor:	Hostess, Waiter, Manager, Chef
Actor's Goal:	To successfully send either the time the actor begins or ends working a shift
Participating Actors:	
Preconditions:	User has logged in successfully: Include Login (UC-1)
Postconditions:	User successfully entered hours
Failed End Condition:	User cancels EnterHours process

Flow of Events for Main Success Scenario:

→ 1. User selects Enter Hours option
← 2. System prompts user to Clock-In or to Clock-Out
→ 3. User selects desired option
← 4. (a) System receives UserName, Date, and Time of selected option (b) Notifies user of successful time punch (c) returns to main interface

Flow of Events for Extensions (Alternate Scenarios):

3 a. User selects Clock-In
  b. User selects Clock-Out
  c. User selects Cancel
    ← 1. (a) System Cancels request to EnterHours (b) returns to main interface

Use Case UC-3	**ManageEmployees**
Related Requirements:	REQ2
Initiating Actor:	Manager
Actor’s Goal:	To successfully view a list of all current employees with the option of adding, removing or editing an employee
Participating Actors:	Waiter
Preconditions:	User is authorized as a Manager and has logged in successfully: Include Login (UC-1)
Postconditions:	Employee list has been successfully updated in the system
Failed End Condition:	Employee list failed to updated
Flow of Events for Main Success Scenario: → 1. Manager selects Manage Employees option from main interface ← 2. System displays a list of current employees with selection options (add,edit,delete) → 3. Manager selects desired option ← 4. System successfully updates the Employee List and notifies Manager of success	
Flow of Events for Extensions (Alternate Scenarios): 3 a. Manager selects Add Employee: Include AddEmployee (UC-4) b. Manager selects Edit Employee: Include EditEmployee (UC-6) c. Manager selects Delete Employee: Include RemoveEmployee (UC-5) d. Manager selects Cancel ← 1. System returns Manager to main interface	

Use Case UC-4	**AddEmployee**
Related Requirements:	REQ2
Initiating Actor:	Manager
Actor’s Goal:	To successfully add a new employee to Employee List
Participating Actors:	
Preconditions:	User is authorized as Manager and has logged in successfully: Include Login (UC-1)
Postconditions:	System successfully updated Employee List
Failed End Condition:	Employee List failed to update

Flow of Events for Main Success Scenario:

→ 1. Manager selects Add Employee option from ManageEmployees interface

← 2. System prompts Manager to fill out text fields for employee information (First Name, Last Name, Phone Number, House Number, Street, City, State, Zip Code, SSN, Wage)

→ 3. Manager enters all information

← 4. (a) System validates new employee information (b) System assigns a unique identification number to new employee(c) updates Employee List

Flow of Events for Extensions (Alternate Scenarios):

3 a. Manager selects Cancel option
    ← 1. (a) System does not add employee (b) returns to ManageEmployees Interface
4 a. System detects non-letter element in First Name, Last Name, City, State
    ← 1. System notifies Manager that data field is invalid or missing
  b. System detects a letter element in Phone Number, Wage, House Number, Zip Code, SSN
    ← 1. System notifies Manager that data field is invalid or missing
  c. System detects an empty data field
    ← 1. System notifies Manager that data field is invalid or missing

Use Case UC-5	**RemoveEmployee**
Related Requirements:	REQ2
Initiating Actor:	Manager
Actor's Goal:	To successfully remove an employee from the Employee List
Participating Actors:	
Preconditions:	User is authorized as Manager and has logged in successfully: Include Login (UC-1) Removed Employee is not currently logged into the system Removed Employee is has already clocked out Removed Employee has returned all company belongings
Postconditions:	System successfully updated Employee List
Failed End Condition:	Employee List failed to update

Flow of Events for Main Success Scenario:

→ 1. Manager selects Delete option next to employee's name
← 2. System asks Manage if the employee selected should be deleted from list
→ 3. Manager confirms action to delete employee
→ 4. (a) System confirms employee has been paid
(b) System removes employee from Employee List (b) updates Employee List

Flow of Events for Extensions (Alternate Scenarios):

3 a. Manager selects Cancel option
← 1. System does not delete employee
4 (a). System alerts Manager that employee has not been paid
System does not remove employee from employee roster
System aborts RemoveEmployee and opens Payroll *<<include>> ViewPayroll (UC-13)*

Use Case UC-6	EditEmployee
Related Requirements:	REQ2
Initiating Actor:	Manager
Actor’s Goal:	To successfully edit an employee’s information
Participating Actors:	
Preconditions:	User is authorized as Manager and has logged in successfully: Include Login (UC-1)
Postconditions:	System successfully updated Employee List
Failed End Condition:	Employee List failed to update

Flow of Events for Main Success Scenario:

→ 1. Manager selects Edit option next to desired employee’s name
← 2. System displays text fields (First Name, Last Name, Phone Number, House Number, Street, City, State, Zip Code, Wage) of employee’s information
→ 3. (a)Manager edits desired text fields (b) Selects Update Info option
← 4. (a) System validates new information (b) updates Employee List

Flow of Events for Extensions (Alternate Scenarios):

3 a. Manager selects Cancel option
    ← 1. (a) System does not edit employee (b) returns to ManageEmployees Interface
4 a. System detects non-letter element in First Name, Last Name, City, State
    ← 1. System notifies Manager that data field is invalid or missing
  b. System detects a letter element in Phone Number, Wage, House Number, Zip Code
    ← 1. System notifies Manager that data field is invalid or missing
  c. System detects an empty data field
    ← 1. System notifies Manager that data field is invalid or missing

Use Case UC-7	**ManageMenu**
Related Requirements:	REQ9
Initiating Actor:	Manager
Actor's Goal:	To successfully add, edit or remove an item from the menu
Participating Actors:	
Preconditions:	User is authorized as a Manager and has logged in successfully: Include Login (UC-1) Manager alters menu while restaurant is not in operation (before opening and after closing)
Postconditions:	System successfully updated Menu
Failed End Condition:	Menu failed to update

Flow of Events for Main Success Scenario:

→ 1. Manager selects Manage Menu option from the main interface
← 2. System displays ManageMenu Interface with options to Add,Edit,Remove,Cancel
→ 3. Manager selects desired option
← 4. System successfully updates menu (b) notifies manager of success

Flow of Events for Extensions (Alternate Scenarios):

3 a. Manager selects Add Employee: Include AddMenuItem (UC-8)
  b. Manager selects Edit Employee: Include EditMenuItem (UC-9)
  c. Manager selects Delete Employee: Include DeleteMenuItem (UC-10)
  d. Manager selects Cancel
    ← 1. System returns Manager to main interface

Use Case UC-8	**AddMenuItem**
Related Requirements:	REQ9
Initiating Actor:	Manager
Actor's Goal:	To successfully add a new menu item to Menu
Participating Actors:	
Preconditions:	User is authorized as Manager and has logged in successfully: Include Login (UC-1) Manager alters menu while restaurant is not in operation (before opening and after closing)
Postconditions:	System successfully updated Menu
Failed End Condition:	Menu failed to update

Flow of Events for Main Success Scenario:

→ 1. Manager selects Add Menu Item option from ManageMenu interface

← 2. System prompts Manager to fill out information required for creating a new menu item (Name, description, price, Side Dishes)

→ 3. Manager enters all information

← 4. (a) System validates new menu item information (b) updates Menu

Flow of Events for Extensions (Alternate Scenarios):

3 a. Manager selects Cancel option

    ← 1. (a) System does not add menu item (b) returns to ManageMenu Interface

4 a. System detects a letter element in price data field

    ← 1. System notifies Manager that data field is invalid or missing

  b. System detects a data field is empty

    ← 1. System notifies Manager that data field is invalid or missing

Use Case UC-9	RemoveMenuItem
Related Requirements:	REQ2
Initiating Actor:	Manager
Actor’s Goal:	To successfully remove an menu item from Menu
Participating Actors:	
Preconditions:	User is authorized as Manager and has logged in successfully: Include Login (UC-1) Manager alters menu while restaurant is not in operation (before opening and after closing)
Postconditions:	System successfully updated Menu
Failed End Condition:	Menu failed to update

Flow of Events for Main Success Scenario:

→ 1. Manager selects Delete option next to menu item’s name
← 2. System asks Manage if the menu item selected should be deleted from list
→ 3. Manager confirms action to delete menu item
← 4. (a) System removes menu item from Menu (b) updates Menu

Flow of Events for Extensions (Alternate Scenarios):

3 a. Manager selects Cancel option
    ← 1. System does not delete menu item

Use Case UC-10	**EditMenuItem**
Related Requirements:	REQ2
Initiating Actor:	Manager
Actor’s Goal:	To successfully edit a menu item’s information
Participating Actors:	
Preconditions:	User is authorized as Manager and has logged in successfully: Include Login (UC-1) Manager alters menu while restaurant is not in operation (before opening and after closing)
Postconditions:	System successfully updated Menu
Failed End Condition:	Menu failed to update

Flow of Events for Main Success Scenario:

→ 1. Manager selects Edit option next to desired menu item
← 2. System displays text fields of menu item’s information
→ 3. (a)Manager edits desired text fields (b) Selects Update Info option
← 4. (a) System validates new information (b) updates Menu

Flow of Events for Extensions (Alternate Scenarios):

3 a. Manager selects Cancel option
    ← 1. (a) System does not edit menu item (b) returns to ManageMenu Interface

Use Case UC-11	ManageLayout
Related Requirements:	REQ5
Initiating Actor:	Manager
Actor's Goal:	To successfully edit the floor plan of the restaurant
Participating Actors:	Waiter
Preconditions:	User is authorized as a Manager and has logged in successfully: Include Login (UC-1)
Postconditions:	System successfully updated floor plan of restaurant
Failed End Condition:	floor plan failed to update

Flow of Events for Main Success Scenario:

→ 1. Manager selects Floor Plan option from main interface
← 2. System displays a grid representation of the floor plan and options
→ 3. (a) Manager clicks on desired table (b) moves table to new location
← 4. (a) System verifies location of table (b) updates floor plan

Flow of Events for Extensions (Alternate Scenarios):

3 a. Manager selects Add new Table option
    ← 1. (a) System creates a new table (b) places table as close as possible to center of grid
  b. Manager long clicks on a table
    ← 1. System prompts user to edit number of seats at the table, delete table, cancel
    → 2. Manager selects desired option to edit floor plan
    ← 3. (a) System verifies changes (b) updates floor plan
  c. Manager selects Back to Main Menu option
    ← 1. System returns Manager back to main interface

Use Case UC-12	**ManageAds**
Related Requirements:	REQ9
Initiating Actor:	Manager
Actor's Goal:	To successfully edit the advertisements displayed while Menu Application is idle
Participating Actors:	
Preconditions:	User is authorized as a Manager and has logged in successfully: Include Login (UC-1)
Postconditions:	System successfully updated advertisement slide show
Failed End Condition:	Advertisement Slide Show failed to update

Flow of Events for Main Success Scenario:

→ 1. Manager selects Manage Ads option from Main Interface
← 2. System displays a slide show of advertisement images and options Next,Prev,Add,Delete,Return
→ 3. Manager selects desired option
← 4. System successfully updates Advertisement List

Flow of Events for Extensions (Alternate Scenarios):

3 a. Manager selects Next option
    ← 1. System displays next Advertisement image in Advertisement List
  b. Manager selects Prev option
    ← 1. System displays previous Advertisement image in Advertisement List
  c. Manager selects Add option
    ← 1. System prompts Manager for PATH of new advertisement
    → 2. Manager provides PATH of desired Advertisement Image
    ← 3. (a) System verifies image PATH (b) updates Advertisement List
  d. Manager selects Delete option
    ← 1. System asks if image should be removed from Advertisement List
    → 2. Manager confirms deletion by selecting OK option
    ← 3. (a) System removes image from Advertisement List (b) notifies Manager of the update

Use Case UC-13	**ViewPayroll**
Related Requirements:	REQ7
Initiating Actor:	Manager
Actor's Goal:	To successfully pay an employee
Participating Actors:	
Preconditions:	User is authorized as a Manager and has logged in successfully: Include Login (UC-1)
Postconditions:	System updates payroll table
Failed End Condition:	System fails to update payroll

Flow of Events for Main Success Scenario:

→ 1. Manager selects Payroll option from main interface
← 2. (a)System displays table with columns: Employee Name, Biweekly Hours Logged, Paycheck Owed, Paid Yet, Total Hours Logged, Total Paycheck
    (b) displays options next to each employee name to: Pay Employee, Edit Hours
    (c) displays option to return to main menu
→ 3. Manager views Payroll and selects desired options
← 4. System updates Payroll information and notifies manager of success

Flow of Events for Extensions (Alternate Scenarios):

3 a. Manager selects Pay Employee option
    ← 1. System asks to confirm payment of the Manager Selected Employee
    → 2. Manager confirms payment
    ← 3. System prints out paycheck
  b. Manager selects Edit Hours option
    ← 1. System displays Biweekly Hours Logged of the Manager Selected Employee and prompts Manager to enter new value
    → 2. Manager enter new value for Biweekly Hours
    ← 3. System notifies Manager of success
  c. Manager selects Back to Main Menu option
    ← 1. System returns Manager back to main interface

Use Case UC-14	**ViewSales**
Related Requirements:	REQ7
Initiating Actor:	Manager
Actor’s Goal:	To successfully display a report of all sales made by the restaurant
Participating Actors:	
Preconditions:	User is authorized as a Manager and has logged in successfully: Include Login (UC-1)
Postconditions:	System successfully displays sales report
Failed End Condition:	System fails to display sales report

Flow of Events for Main Success Scenario:

→ 1. Manager selects Sales Report option from main interface
← 2. (a)System displays date of report with options to go to previous/next date, change time span of report (daily, weekly, monthly, etc.)
    (b) displays a table with columns: Menu Item, Menu Item Price, Revenue Menu Item
    (c) displays option to print report, return to main menu
→ 3. Manager views Sales Report and selects desired option
← 4. System updates Sales Report

Flow of Events for Extensions (Alternate Scenarios):

3 a. Manager Change Time Span option
    ← 1. System displays a drop down menu of different time spans
    → 2. Manager selects desired timespan
    ← 3. System displays Sales Report over new time span
  b. Manager selects Next option
    ← 1. (a) System verifies data was collected for the Next Date (b) displays Sales Report for Next Date
  c. Manger selects Previous option
    ← 1. (a) System verifies data was collected for the Previous Date (b) displays Sales Report for Previous Date
  d. Manager selects Back to Main Menu option
    ← 1. System returns Manager back to main interface

Use Case UC-15	**TableStatus**
Related Requirements:	REQ5
Initiating Actor:	Hostess
Actor's Goal:	To view and assign tables based on availability
Participating Actors:	Waiter
Preconditions:	Customer enters the restaurant ready to be seated
Postconditions:	Customer is seated and ready to place an order
Failed End Condition:	Floor plan indicates non-availability of a table

Flow of Events for Main Success Scenario:

← 1. System will let Hostess log in to the system if he/she is already not logged in:
<<include>> "Login (UC-1)"
→ 2. Hostess displays floor "Restaurant Floor Plan" option on the system for availability status
← 3. System (a) lets Hostess check the floor plan for availability
(b) displays the restaurant's floor layout with tables marked as
"Ready", "Occupied", "Reserved" or "Dirty"
→ 4. Hostess selects a "Ready" table to be assigned to Waiter
← 5. System assigns Waiter to table
→ 6. Waiter is notified by the System and greets Customer to the assigned table
→ 7. Customer is ready to place an order
← 8. System will let Waiter change the table status to "Occupied"

Flow of Events for Extensions (Alternate Scenarios):

3(b)
← 1. System displays all tables as "Occupied"
← 2. System determines wait time
← 3. Hostess informs customer about wait time and requests Customer to wait
→ 4. Wait time is not acceptable to Customer and Customer leaves; use case ends here

3(b)
→ 1. Customer comes in with a reservation
← 2. System displays "Reserved" tables
← 3. System lets Hostess confirm Customer's reservation information (name and phone number)
→ 4. Hostess selects Customer's "Reserved" table to be assigned to Waiter
Follow steps 5-8 from Main Success Scenario

Use Case UC-16	**OrderFood**
Related Requirements:	REQ3
Initiating Actor:	Customer
Actor's Goal:	To select desired meal items from the food menu
Participating Actors:	
Preconditions:	Customer is seated and ready to order
Postconditions:	Customer has placed the order
Failed End Condition:	Item ordered by customer is not available
Flow of Events for Main Success Scenario: → 1. Customer selects "View Menu" from the main screen ← 2. System displays a list of food items available to Customer → 3. Customer selects food items of his/her choice ← 4. System confirms the order and reorganizes the food items based on food priority and sends it to the Chef's *Order Queue* ← 5. System sends a list or Customer's ordered items to the Waiter	

Use Case UC-17	**CancelOrder**
Related Requirements:	REQ3
Initiating Actor:	Customer
Actor's Goal:	To cancel a food order
Participating Actors:	Waiter
Preconditions:	Customer has placed a food order Cancelled Item is in the *Order Queue*
Postconditions:	Food item is removed from the *Order Queue*
Failed End Condition:	Food item has already been made by Chef

Flow of Events for Main Success Scenario:

→ 1. Customer selects "Call Waiter" from menu: <<include>> "Call Waiter" UC-18
→ 2. Customer tells waiter which food item to cancel
→ 3. Waiter logs into the system if not already logged in:
<<include>> "Login (UC-1)"
→ 4. Waiter clicks on desired table
← 5. System displays all food items ordered by that table and the food status
→ 6. Waiter selects the food item to be removed from the *Order Queue*
← 7. System asks Waiter to confirm
→ 8. Waiter presses "OK"
← 9. System updates the *Order Queue* and notifies Chef that the item has been removed

Flow of Events for Extensions (Alternate Scenarios):

5. Cancelled Item Request is already in *Processing Queue* (Food Status says 'Began Cooking", "5 min", "10 min", "Done")
→ 1) Waiter notifies Customer that the request cannot be made

5. Cancelled Item Request is already in *Processing Queue* (Food Status says 'Began Cooking", "5 min", "10 min", "Done")
→ 1) Waiter notifies Customer that the request cannot be made
→ 2) Customer demands item to be removed
→ 3) Waiter notifies Manager of cancellation override request
→ 4) Manager enters Override Code to delete item from Waiter Interface
← 5) System removes item from Order, Bill, and Processing Queue (if item is still cooking)

8.
→ 1. Waiter presses "NO"
Follow steps 5-9 from Main Success Scenario

Use Case UC-18	**ServeFood**
Related Requirements:	REQ8
Initiating Actor:	Chef
Actor's Goal:	To allow Waiter to serve food to Customer after Chef updates status to "Ready"
Participating Actors:	Waiter
Preconditions:	Chef receives food order submitted by each table
Postconditions:	Order is delivered to Customer's Table
Failed End Condition:	None

Flow of Events for Main Success Scenario:

→ 1. (a) Chef selects "View Orders" option from main screen
    (b) Chef selects the food order marked with highest priority from *Order Queue*
    (c) The selected order is moved into the Processing Queue
← 2. System updates the food status to "Being Processed"
→ 3. Chef changes food status to "Ready" once the meal is prepared
← 4. System notifies waiter of food status once it is ready for pickup
→ 5. Waiter logs into system if not already logged in *<<include>> "Login (UC-1)"*
→ 6. Waiter (a) picks up food order from kitchen
        (b) serves the Customer
        (c) changes food status to "Served"
← 7. System verifies and updates food status

Use Case UC-19	**CallWaiter**
Related Requirements:	REQ6
Initiating Actor:	Customer
Actor’s Goal:	To allow Customer to call Waiter for assistance
Participating Actors:	Waiter
Preconditions:	Customer is seated at a table
Postconditions:	System has notified Waiter of the Customer’s call
Failed End Condition:	

Flow of Events for Main Success Scenario:

→ 1. Customer selects “Call Waiter” option from the menu’s main screen
← 2. System receives “Call Waiter” notification and directs the alert to the assigned waiter at that table
→3. Waiter logs into the system if not already logged in: *<<include>> “Login (UC-1)”*
← 6. System allows Waiter to view which table requires assistance
→ 7. Waiter proceeds to that table to assist Customer

Use Case UC-20	**PayBill**
Related Requirements:	REQ6 REQ10
Initiating Actor:	Customer
Actor’s Goal:	To pay for their meals when Customer is ready to leave
Participating Actors:	Waiter, Manager
Preconditions:	Every item ordered by Customer was served
Postconditions:	Customer receives receipt after bill payment
Failed End Condition:	

Flow of Events for Main Success Scenario:

→ 1. Customer selects “View Bill” option from the main screen
← 2. System displays a list of Customer’s ordered meals along with each meal price
→ 3. Customer selects “Tip Calculator” to include tip in overall total
← 4. System calculates the total cost
→ 5. Customer presses “Call Waiter”: *<<include>> “Call Waiter (UC-19)”*
→ 7. (a) Waiter collects payment for bill from Customer
    (b) Waiter returns to terminal and logs into System: *<<include>>”Login (UC-1)”*
    (c) Waiter selects proper payment plan
← 8. System verifies that bill payment has been made

Flow of Events for Extensions (Alternate Scenarios):

7(c)
→ 1. Waiter selects “Cash Payment” *<<include>> “CashPayment (UC-21)”*
  2. Waiter selects “Card Payment” *<<include>> “CardPayment (UC-22)”*

Use Case UC-21	**CashPayment**
Related Requirements:	REQ10
Initiating Actor:	Waiter
Actor's Goal:	To pay Customer's bill and update sales record
Participating Actors:	
Preconditions:	Customer notifies Waiter of the desired bill payment
Postconditions:	Customer receives receipt after bill payment
Failed End Condition:	

Flow of Events for Main Success Scenario:

→ 1. Waiter selects "Pay Bill" option from main screen
← 2. System displays two payment options: (i) "Cash Payment" (ii) "Card Payment"
→ 3. Waiter (a) selects "Cash Payment"
(b) selects table number requesting for bill payment
(c) selects "View Bill" for the designated table
← 4. System displays a list of Customer's ordered meals along with the total cost
→ 5. Waiter processes bill payment by cash and selects "Print Receipt"
← 6. System (a) verifies that bill payment has been made
(b) updates table status to "Dirty"

Use Case UC-22	**CardPayment**
Related Requirements:	REQ10
Initiating Actor:	Waiter
Actor’s Goal:	To pay Customer’s bill and update sales record
Participating Actors:	
Preconditions:	Customer notifies Waiter of the desired bill payment
Postconditions:	Customer receives receipt after bill payment
Failed End Condition:	

Flow of Events for Main Success Scenario:

→ 1. Waiter selects “Pay Bill” option from main screen
← 2. System displays two payment options: (i) “Cash Payment” (ii) “Card Payment”
→ 3. Waiter (a) selects “Card Payment”
    (b) selects table number requesting for bill payment
    (c) selects “View Bill” for the designated table
← 4. System displays a list of Customer’s ordered meals along with the total cost
→ 5. Waiter processes bill payment by swiping Customer’s credit card and selects “Print Receipt”
← 6. System (a) verifies that bill payment has been made
    (b) updates table status to “Dirty”

Use Case UC-23	**SendReservation**
Related Requirements:	REQ5
Initiating Actor:	Customer
Actor’s Goal:	To make a reservation at a desired time
Participating Actors:	
Preconditions:	None
Postconditions:	Customer’s reservation has been created for a specific date and time
Failed End Condition:	A reservation has not been made

Flow of Events for Main Success Scenario:

→ 1. Customer selects “Place Reservation” from main screen
← 2. System displays available dates (next three days including current day) and timings for reservation
→ 3. Customer selects reservation date and time of preference and clicks “Continue”
← 4. System displays a new screen with input fields for: name, phone number, and email address
→ 5. Customer enters valid information into the fields and clicks “Submit”
← 6. System verifies the reservation request and displays an automated message saying “Reservation Successfully Made” and provides Customer with a unique reservation number

Flow of Events for Extensions (Alternate Scenarios):

2.
← 1. System displays “Sorry, no available dates and timings for the next three days”
→ 2. Customer clicks “Exit” to exit Android application

5 a. System detects a non-letter element in Name
    ← 1. System notifies Manager that data field is invalid or missing
  b. System detects a letter element in Phone Number
    ← 1. System notifies Manager that data field is invalid or missing
  b. System detects a data field is empty
    ← 1. System notifies Manager that data field is invalid or

Use Case UC-24	**CancelReservation**
Related Requirements:	REQ5
Initiating Actor:	Customer
Actor's Goal:	To cancel a reservation
Participating Actors:	
Preconditions:	Reservation has already been made.
Postconditions:	The selected reservation has been cancelled
Failed End Condition:	The selected reservation has not been cancelled

Flow of Events for Main Success Scenario:

→ 1. Customer selects “Cancel Reservation” from main screen
← 2. System displays a screen with an input field for a reservation number
→ 3. Customer enters a reservation number and clicks “Continue”
← 4. System displays a new screen with the details of the reservation
→ 5. Customer selects “Process Cancellation” from the screen
← 6. System verifies that reservation request has been cancelled and displays an automated message saying “Reservation Successfully Cancelled”

Flow of Events for Extensions (Alternate Scenarios):

3.
→ 1. Customer (a) enters invalid reservation number into the fields
← 2. System redisplays the input field and a message stating that required data is invalid or missing
Follow steps 4-6 from Main Success Scenario

Traceability Matrix

Req’t	PW	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	5	X																							
2	5			X	X	X	X			X	X														
3	5																X	X							
4	2																								
5	3											X				X								X	X
6	5																			X	X				
7	5		X											X	X										
8	1																		X						
9	2							X	X				X												
10	4																				X	X	X		
Max PW		5	5	5	5	5	5	2	2	5	5	3	2	5	5	3	5	5	1	5	5	4	4	3	3
Total PW		5	5	5	5	5	5	2	2	5	5	3	2	5	5	3	5	5	1	5	9	4	4	3	3

User Interface Design

CUSTOMER INTERFACE

The Customer will be in communication with the System through an Android Tablet Application representing the restaurant menu. Upon seating the Customers, the Hostess will provide the table with an Android Tablet displaying the Menu Home Page (Image #1). The Menu Home Page provides everything for the Customers to best enjoy their dining experience including options to Call the Waiter, Order Food, View the Order, and Pay the Bill.

Once the Order Option is selected, the Android Tablet displays the Menu Interface (Image #2) which lists all Menu Items organized by categories such as Appetizers, Burgers, Desserts, etc. All categories are presented as Buttons in the top-left corner of the screen. Because of sizing issues, only six Category Buttons are presented at a time. The first five buttons will be Menu Categories and the sixth button will be a 'More' Option that displays the remaining Menu Categories if there are More than six.

After selecting the desired category, all Menu Items within the category are displayed on the left side of the tablet. This list is clickable and will be scrollable if all items do not fit on the screen. The right side of the screen will contain a message prompting the Customer to select a Menu Item from the List. Once an item is selected, the right side of the screen displays everything needed for a user to fully customize any order. The Customer will see a description of the Menu Item selected, including certain toppings or ingredients as well as the price of the item. Beneath the description is a text box for the Customer to leave requests for the Chef about the order. A typical request may be to cook a burger to medium rare, or for an extra sauce or dressing on the side. The Customer will also be able to choose the side to accommodate the meal. These options are presented as radio buttons for the Customer to choose from. Next to the Meal Sides is a button to add the Meal to the Order List. If the Customer does not want to add this order, then he or she can click either a different Menu Category or a different Menu Item and repeat the process. When the Customer adds the Meal to the Order List, the Android Tablet will display a notification of the successful addition and will also say that unwanted items can be removed from the View Order Menu.

The View Order Interface (Image #3) can be selected from the Menu Interface by selecting the button in the top right corner of the screen. The Menu Interface also provides options for the Customer to call the Waiter for help and to also cancel the entire Order. The Cancel Order option will warn the Customer of the action before committing. When the Customer has finished selecting from the Menu, the View Order interface will display a scrollable list of all items ordered. Next to each item will be a button to remove that item of the Order List. The Customer will be warned once again before the Android Application commits to the deletion. The View Order Interface will provide options for the Customer to Call the Waiter, to Cancel the Order, to Send the Order, and to Add to the Order. Selecting Add to Order will return the Customer to the Menu Interface and the above steps can be repeated. The Customer can finalize the order by sending it to the chef through the Send Order Option.

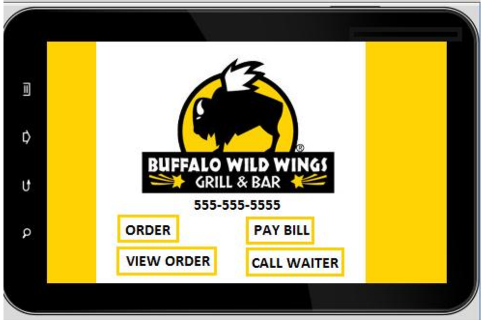
After selecting the Send Order Option the Customer will be returned to the Main Interface where the user can once again Order, View Order, Pay Bill, and Call Waiter. If the Customer attempts to View the Order, the same View Order Interface will appear once again but items already sent to the chef will not have Delete Buttons next to them. A waiter will need to override any items sent to the chef that are no longer desired.

The Customer will also be able to pay the bill from the Main Interface by selecting the Pay Bill Option. Selecting so will present the Customer with the Pay Bill Interface. The screen will display a scrollable List of all items ordered. Each item in the List will display the name of the Menu Item selected, the selected Side Item, the Note for Chef, and the Price of the item. Beneath the Order List is a Text Box for the Customer to specify a tip for the waiter. The Customer can also use the Easy Tip option located next to the Text Box, which will automatically add a ten, fifteen, eighteen, or twenty percent tip (excluding tax). Next to the Tip Options, the Bill Total will be presented. This Total will be the sum of all items ordered, all drinks, tax, and tip. The Customer cannot pay from the Tablet and must call the Waiter who will accept the payment and return a receipt and change.

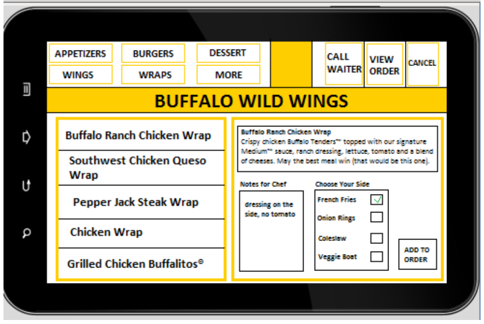
Any time the Android Application is idle for 3 minutes, the Customer will see the Advertisement Interface. From here, the Customer will see a slide show of restaurant related images (food, drinks, logos, etc.) selected by the Manager. These images will change every 30 seconds until the Customer touches the picture. There will also be arrow options for the Customer to manually change the current image of the slide show. Clicking the image will return the Customer to the Main Interface.

Figure X displays a flow chart of how all Android Interfaces Interact. Several things are worth noticing: The Pay Bill Interface can only be reached from the Main Interface. This is because the Customer has no need of paying until an order has been sent. It is also important to note that the Customer cannot return to the Main Interface once the Order Process has begun until either the Customer cancels the order or sends it to the chef. Lastly, the Main Interface always has options to View Order and Pay Bill, even before an initial order has been placed. The Android Application will handle empty Orders by displaying a note on the screen saying no orders have been placed.

Main Menu Interface(1)



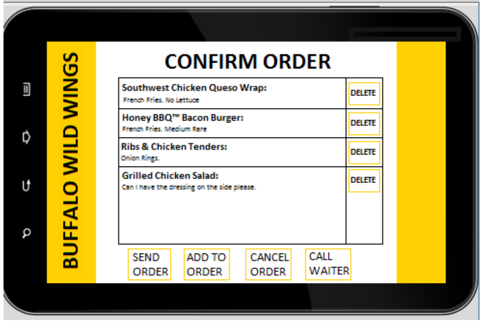
Menu Interface(2)



Buffalo Ranch Chicken Wrap
Crispy chicken Buffalo Tenders™ topped with our signature Medium™ sauce, ranch dressing, lettuce, tomato and a blend of cheeses. May the best meal win (that would be this one).

Notes for Chef: dressing on the side, no tomato

View Order Interface(3)



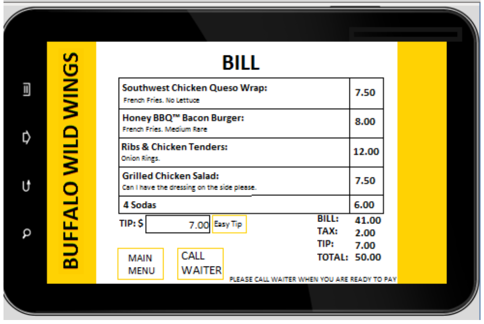
Southwest Chicken Queso Wrap:
French Fries. No Lettuce

Honey BBQ™ Bacon Burger:
French Fries. Medium Rare

Ribs & Chicken Tenders:
Onion Rings.

Grilled Chicken Salad:
Can I have the dressing on the side please.

Pay Bill Interface(4)

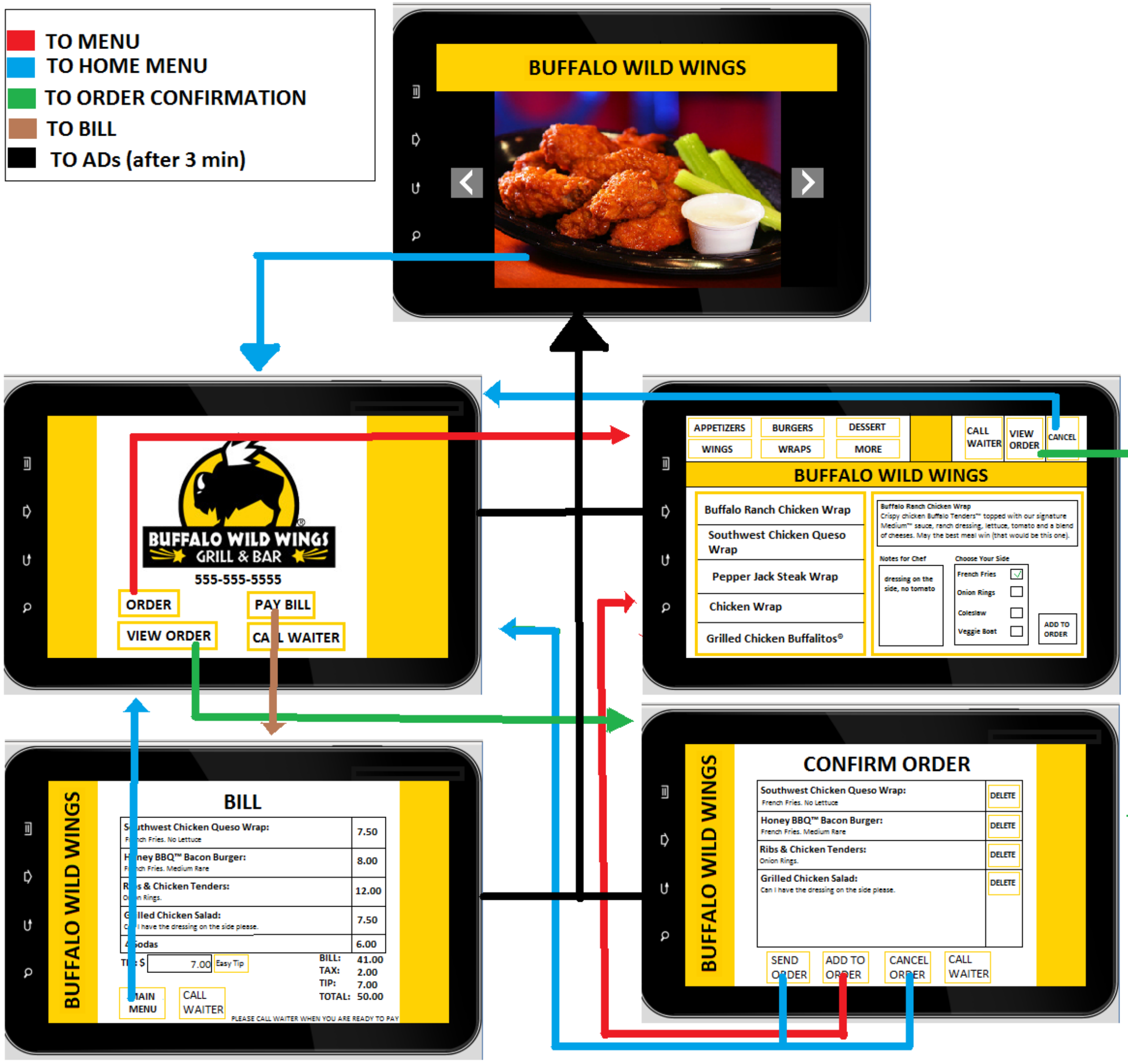


Southwest Chicken Queso Wrap: French Fries. No Lettuce	7.50
Honey BBQ™ Bacon Burger: French Fries. Medium Rare	8.00
Ribs & Chicken Tenders: Onion Rings.	12.00
Grilled Chicken Salad: Can I have the dressing on the side please.	7.50
4 Sodas	6.00

TIP: $ 7.00

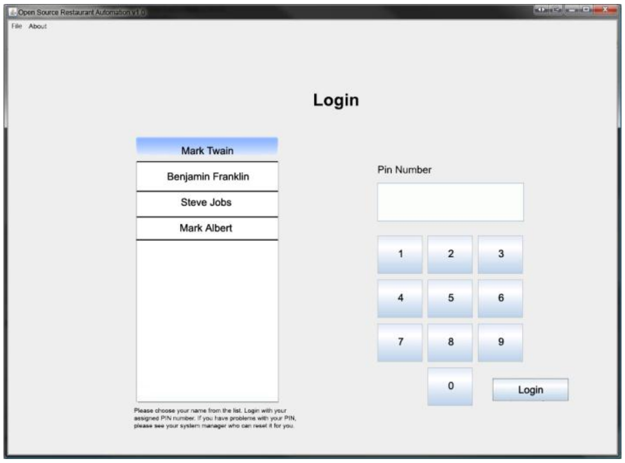
BILL: 41.00
TAX: 2.00
TIP: 7.00
TOTAL: 50.00

PLEASE CALL WAITER WHEN YOU ARE READY TO PAY



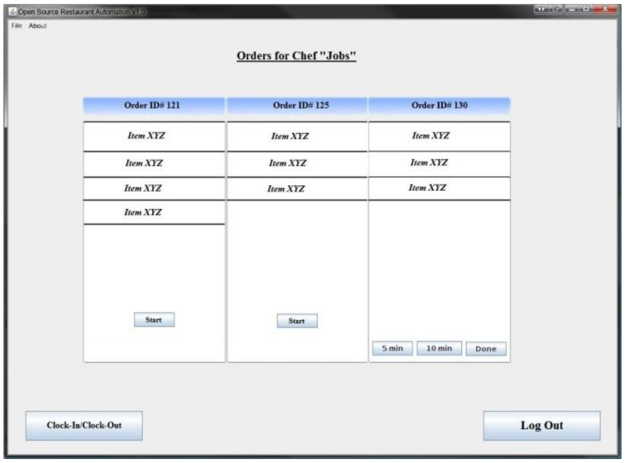
Flow-Chart of Menu Interface:We can see that REQ17 will be addressed through this analysis. REQ25 and REQ22 are also completed through the above interfaces.

Restaurant Personnel Interface:

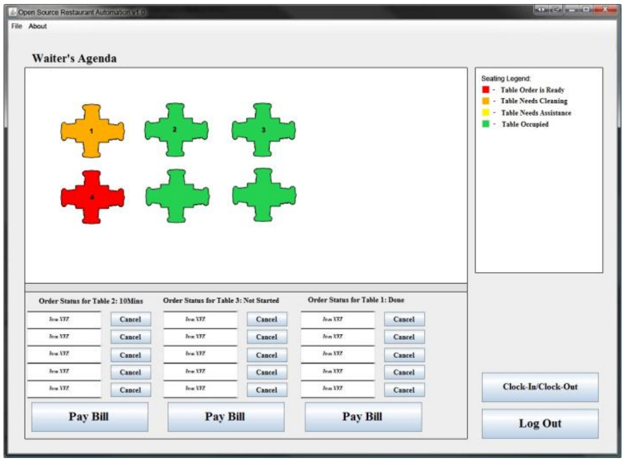


Please choose your name from the list. Login with your assigned PIN number. If you have problems with your PIN, please see your system manager who can reset it for you.

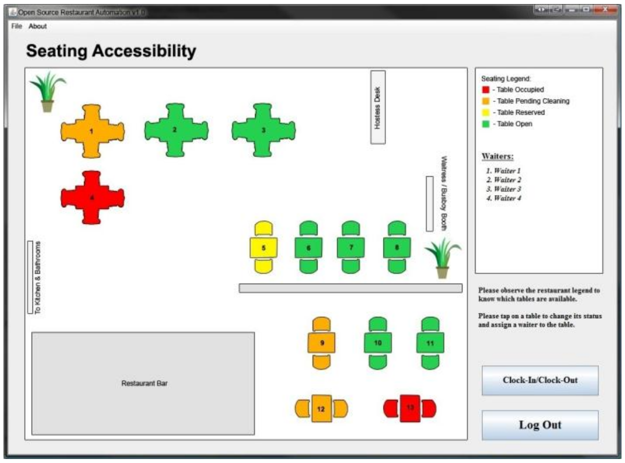
Each individual employee(end-user) in our system will have their name listed on the left side once the manager puts him/her in the system. Once they choose their name they have enter their unique four digit pin and hit login to go their respective interfaces. The figures that follow are the interfaces that will be opening which are determined by the employee's position. This interface addresses REQ15 as it keeps the restaurant's login consistent.



The image above is the interface for the chef. REQ11 and 23 are satisfied based on this interface. The chef will be able to notify the waiter that an order is complete by clicking done. This satisfies requirement 11 as the notification is done through a single screen touch. Appearance Requirement 22 will also be satisfied through this interface as it shows the next 2 orders in queue.


Open Source Restaurant Automation v1.0
File
About
Waiter's Agenda
1
2
3
4
Seating Legend:
🟥 - Table Order is Ready
🟧 - Table Needs Cleaning
🟨 - Table Needs Assistance
🟩 - Table Occupied
Order Status for Table 2: 10Mins
Order Status for Table 3: Not Started
Order Status for Table 1: Done
Item XYZ
Cancel
Pay Bill
Clock-In/Clock-Out
Log Out

The above image is the waiter's screen. REQ24 and REQ20 are satisfied through the above image. The waiter will be able to distinguish between the different services each table requires given that he/she is not color blind and has average eyesight. The waiter can also view chef updates and order details by clicking on the respective tables. The table clicked on most recently appears on the left. The tabs to the right are older tables that the waiter clicked on.



Please observe the restaurant legend to know which tables are available.

Please tap on a table to change its status and assign a waiter to the table.

The above image is the host's screen. Firstly this interface fulfills the host's interface requirement (REQ 21) by showing the exact layout of the restaurant.

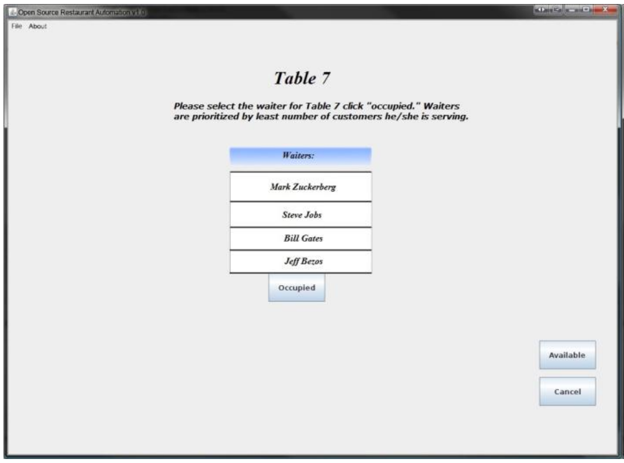


Table 7

Please select the waiter for Table 7 click "occupied." Waiters are prioritized by least number of customers he/she is serving.

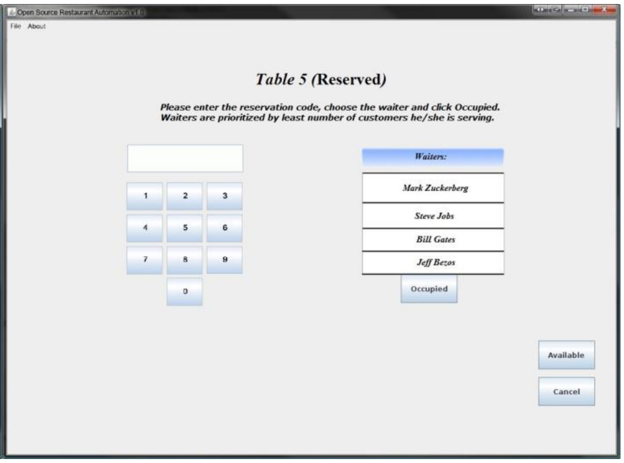
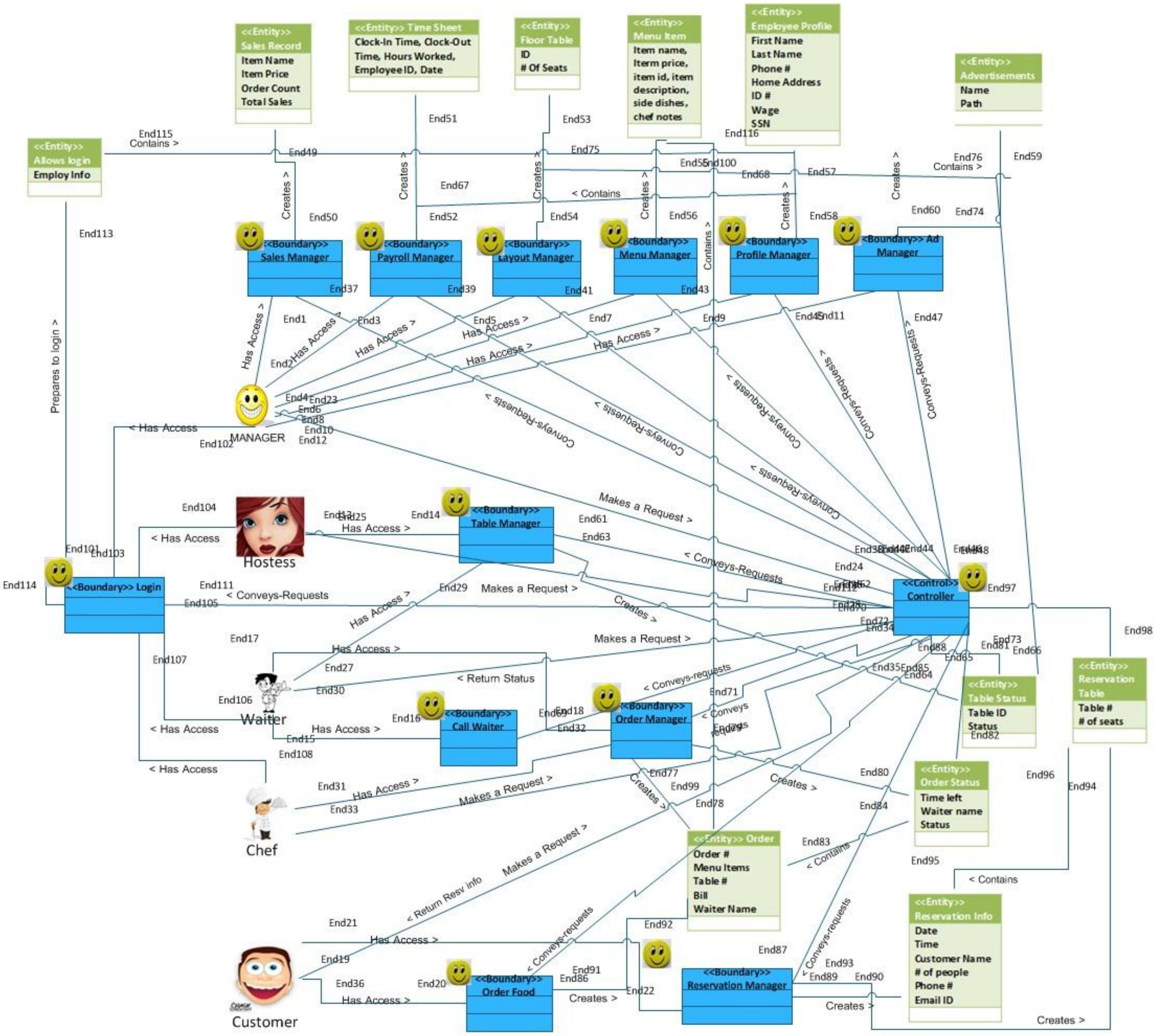


Table 5 (Reserved)

Please enter the reservation code, choose the waiter and click Occupied. Waiters are prioritized by least number of customers he/she is serving.

The host screens above complete REQ18 by predicting which waiter has the least number of customers. The host's screen above also fulfills REQ12 (Screen on right) by prompting for the unique reservation code.

Domain Analysis



Summary

The domain model provides a visual look into the system that is to be designed. Analysis requires attention to detail and explaining how components of the system work together to operate without conflicts. The users have access to specific components (Managers, also known as the domain boundaries [color blue]). The manager has access to the Login, Sales Manager, Payroll Manager, Layout Manager, Menu Manager, and the profile Manager. Manager in this case has access to the major controls of the system. The remaining users (hostess, chef, waiter, and the customer) do not have access to any components accessible by the manager. Hostess is only able to access the table manager since his/her job does not require involvement with other components (boundaries). The waiter can access the call waiter component with limited access to the order manager because the waiter is only required to see the order information along with status updates (waiter does not have an input in the order manager). The chef is only able to control the order manager where he/she can update order status as well as edit customer orders based on item availability. Customer is able to use order food component along with the reservation manager. Customer is able to create a customized order list based on the menu items. They can also make reservations using the reservation manager, which is an automatized component creating or canceling a unique arrangement. Concept definitions provide more details pertaining to the particular function of each component. All accessible components (boundaries) are connected to the controller which allows easy interaction between the user and the system. The system then conveys the special request to a particular component (boundary) made by the user, which then outputs a special entity containing various attributes. The interaction between the user and the controller as well as the controller and the components contain specific associations, which are explained in detail under association definition.

Analysis

Description	Type	Concept Name
Coordinate actions of concepts associated with this use case and delegate the work to other concepts.	D	Controller
Coordinates payroll actions of the manager.	D	Payroll Manager
Coordinates the editing of the menu	D	Menu Manager
Verifies user login information for authorized usage.	D	Login
Coordinates the editing of the floor plan	D	Layout Manager
Manages the editing of advertisements and organizes them to be displayed in slideshow.	D	Ads Manager
Manages the list of current employee information.	D	Profile Manager
Automatically modifies table status to be reserved based on the provided information by the customer.	D	Reservation Manager
Sends the customized customer order to the chef for processing.	D	Order Food
Notifies the waiter of the customer who is in need of assistance	D	Call Waiter
Manages the status of a specific order being processed by the chef (began, 5 min, 10 min, done, etc.)	D	Order Manager
Coordinates the availability status of a specific table (reserved, occupied, etc.)	D	Table Manager
Organizes sales information of each menu item.	D	Sales Manager
Records the employee information	K	Time Sheet
List of all the food items on the menu	K	Menu Item
List of information to display advertisements	K	Advertisements
List of all current employees' ID numbers and personal information	K	Allows Login
List of all revenue earned from a particular menu item	K	Sales Record

Contains each table’s ID number and number of seats	K	Floor Table
Holds each employee’s contact information	K	Employee Profile
Maintains each table’s availability status	K	Table Status
Contains a schedule of all reservations	K	Reservation table
Maintains each table’s food order status	K	Order Status
Contains a list of customer’s desired menu items	K	Order
Holds each customer’s contact information for reserving a table	K	Reservation info

Concept Pair	**Association Description**	**Association Name**
Controller <-> AD Manager	Controller passes requests to AD Manager and displays a list of all Ads on the menu application	Conveys a request
Controller <-> Profile Manager	Controller passes requests to Profile Manager and displays a list of profile of all current employees	Conveys a request
Controller <-> Layout Manager	Controller passes requests to Layout Manager and displays the restaurant’s floor plan	Conveys a request
Controller <-> Menu Manager	Controller passes requests to Menu Manager and displays a list of all menu items	Conveys a request
Controller <-> Payroll Manager	Controller passes requests to Payroll Manager and displays a timesheet	Conveys a request
Controller <-> Sales Manager	Controller passes requests to Sales Manager and displays sales record	Conveys a request
Controller <-> Table Manager	Controller passes requests to Table Manager and displays the status of each table	Conveys a request

Controller <-> Order Manager	Controller passes requests to Order Manager and displays the status of each table’s order	Conveys a request
Controller <-> Call Waiter	Controller passes requests to Call Waiter to alert waiter of a customer in need of assistance	Conveys a request
Controller <-> Reservation Manager	Controller passes requests to Reservation Manager to display a list of reservations made by customers	Conveys a request
Controller <-> Order Food	Controller passes requests to Order Food to send the orders to the Chef	Conveys a request
Controller <-> Login	Controller passes request to Login to verify employee’s unique ID number	Conveys a request
AD Manager <-> Advertisements	Ad manager generates the updated advertisements	Generates/Creates
Profile Manager <-> Employee Profile	Profile manager generates the updated employee information	Generates/Creates
Menu Manager <-> Menu Item	Menu manager creates the updated menu list provided from the manager.	Generates/creates
Layout Manager <-> Floor Table	Generates updated floor plans for the waiter and the hostess.	Generates/Creates
Payroll Manager <-> Timesheet	Generates and saves employee working hours matching them with employee profile.	Generates/Creates
Login <-> Allows Login	Verifies the log information provided by the controller to allow/deny authorized access	Generates/Creates
Sales Manager <-> Sales record	Generates a record of the revenue for analysis for each	Generates/Creates

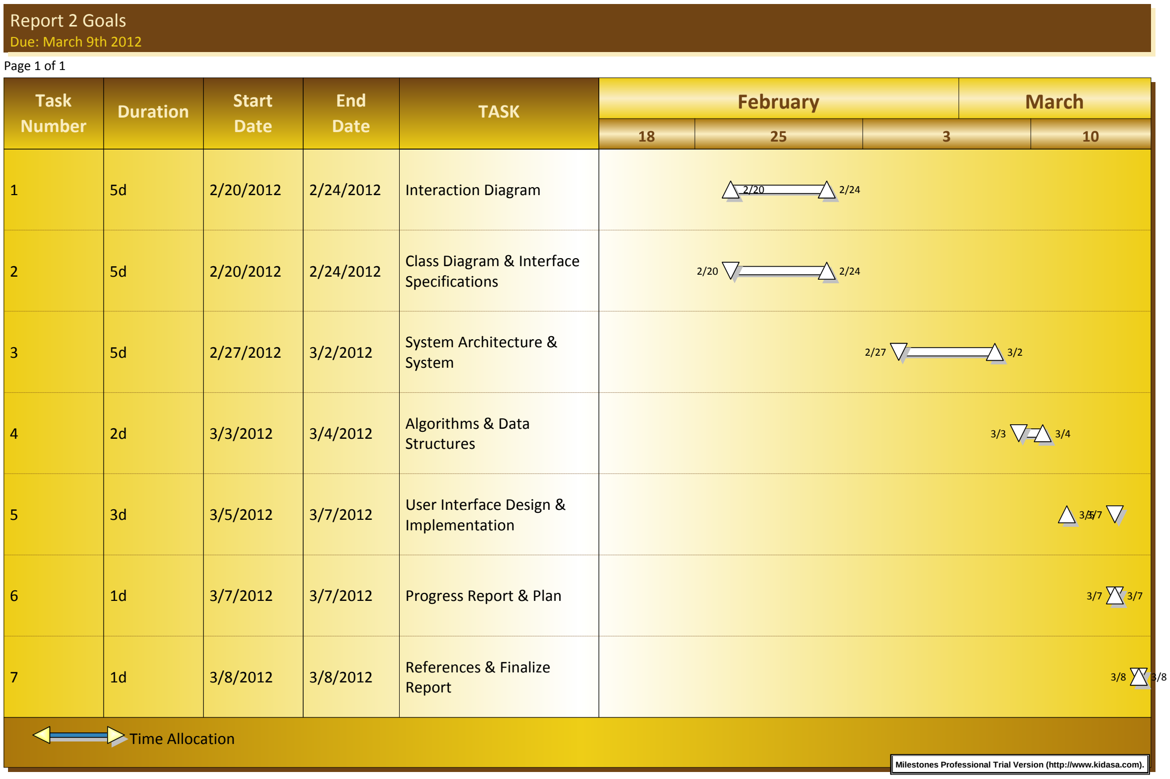
	menu item.	
Table Manager <-> Table Status	Provides the update table availability information by the waiter/hostess.	Generates/Creates
Call Waiter <-> Table Status	Updates the customer assistance status of a particular table.	Generates/Creates
Order Manager <-> Order	Sends the customized order to the chef to begin processing and for the waiter to see.	Generates/Creates
Order manager <-> Order Status	Updates the status of the customized order by the chef (time remaining).	Generates/Creates
Order food <-> Order	Provides the new customized order by the customer.	Generates/Creates
Reservation Manager <-> Reservation Info	Records and generates a customer information list for table reservation.	Generates/Creates

Mathematical Model

Our project does not require any complicated mathematical model. We have a few addition functions in order to calculate wages for employees, restaurant sales, adding menus etc.

Plan of Work

Report 2



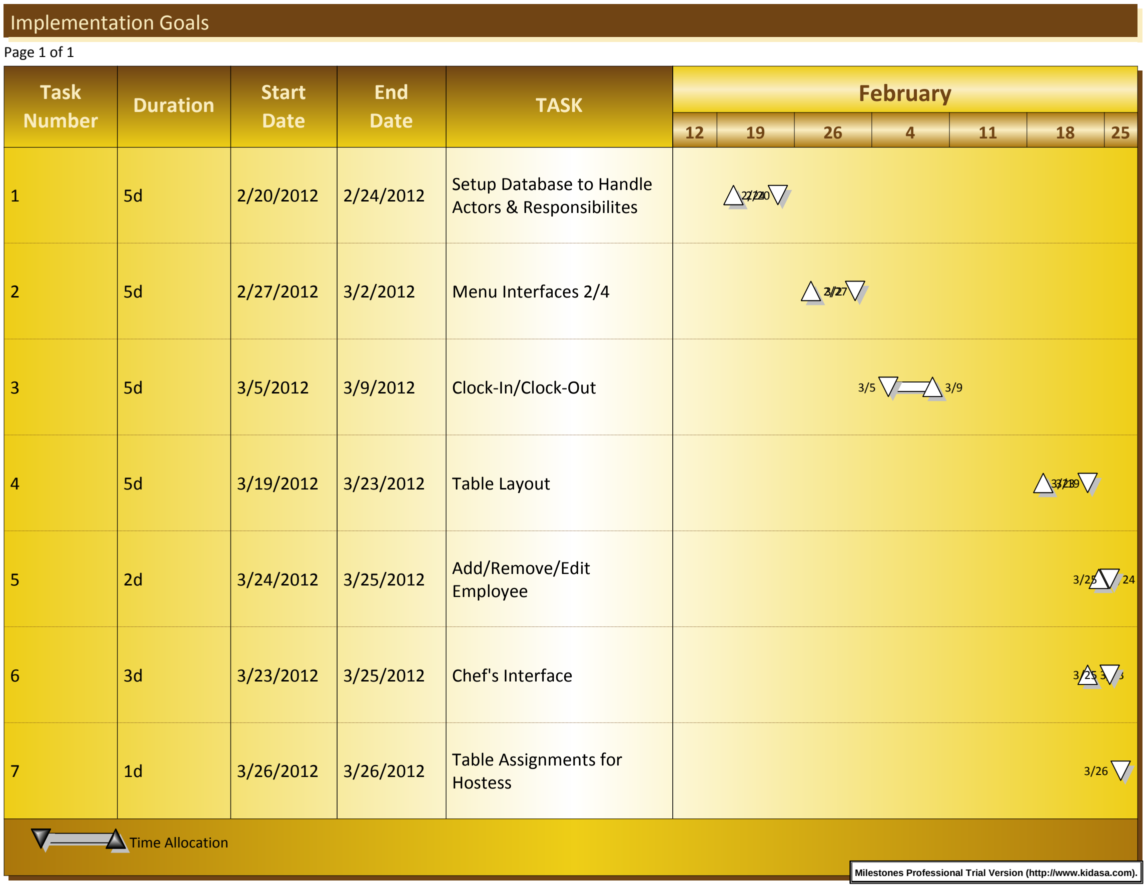
Report 2 Goals

Due: March 9th 2012

Page 1 of 1

Task Number	Duration	Start Date	End Date	TASK
1	5d	2/20/2012	2/24/2012	Interaction Diagram
2	5d	2/20/2012	2/24/2012	Class Diagram & Interface Specifications
3	5d	2/27/2012	3/2/2012	System Architecture & System
4	2d	3/3/2012	3/4/2012	Algorithms & Data Structures
5	3d	3/5/2012	3/7/2012	User Interface Design & Implementation
6	1d	3/7/2012	3/7/2012	Progress Report & Plan
7	1d	3/8/2012	3/8/2012	References & Finalize Report

Implementation



Implementation Goals

Page 1 of 1

Task Number	Duration	Start Date	End Date	TASK
1	5d	2/20/2012	2/24/2012	Setup Database to Handle Actors & Responsibilites
2	5d	2/27/2012	3/2/2012	Menu Interfaces 2/4
3	5d	3/5/2012	3/9/2012	Clock-In/Clock-Out
4	5d	3/19/2012	3/23/2012	Table Layout
5	2d	3/24/2012	3/25/2012	Add/Remove/Edit Employee
6	3d	3/23/2012	3/25/2012	Chef's Interface
7	1d	3/26/2012	3/26/2012	Table Assignments for Hostess

References

Useful Information From:

Book: Software Engineering by Ivan Marsic

Pictures From:

2.)http://www.bostonbakesforbreastcancer.org/on-the-chopping-block/

3.)http://www.chasestockton.com/2010/10/free-vector-cartoon-face/

4.)http://vi.sualize.us/inguyenvu/red%20head/

5.)http://www.principalspage.com/theblog/archives/you-never-forget-your-first-love-especially-if-he-is-a-waiter

6.)http://www.picturesofsmileyfaces.info/

7.)http://www.bww.com/