

---

**Design Report**  
**for**  
**Kitchen Zealot**

**Written by:**  
**Abtahi Chowdhury**  
**Abusaleh Masud**  
**Safwan Shahid**  
**Arman Uddin**  
**Farhan Zaman**

**Version 2.0**

**November 23, 2019**

---

| <b>Date</b> | <b>Version</b> | <b>Description</b>   | <b>Author</b>  |
|-------------|----------------|--|--|
| 11/2/2019   | 1.0.0          | Initial version of the online restaurant system  | Abtahi Chowdhury<br>Abusaleh Masud<br>Safwan Shahid<br>Arman Uddin<br>Farhan Zaman |
| 11/23/2019  | 2.0.0          | This report is meant to provide the data structure and logic to carry out the functionalities dictated by the specification. | Abtahi Chowdhury<br>Abusaleh Masud<br>Safwan Shahid<br>Arman Uddin<br>Farhan Zaman |

# Table of Contents

|  |           |
|--|-----------|
| <b>Introduction</b>                            | <b>5</b>  |
| <b>Design</b>                                  | <b>6</b>  |
| 2.1 Use Case Scenarios                         | 6         |
| 2.2 Collaboration Diagrams                     | 11        |
| 2.3 State Diagrams/ Petri-net                  | 21        |
| <b>E/R Diagram</b>                             | <b>31</b> |
| <b>Pseudocode</b>                              | <b>31</b> |
| addNewEmployee(formValue)                      | 31        |
| login(email:string, password:string)           | 32        |
| register(email:string, password:string)        | 33        |
| logout()                                       | 33        |
| guestLogin()                                   | 33        |
| addToCart(product:Product)                     | 34        |
| removeFromCart(product:Product)                | 34        |
| getCustomer()                                  | 35        |
| getUser()                                      | 35        |
| updateCart(customer:Customer)                  | 35        |
| addToGuestCart(product:Product)                | 36        |
| removeFromGuestCart(product:Product)           | 36        |
| addCustomer(customer:Customer)                 | 37        |
| removeCustomer(customer:Customer)              | 37        |
| updateCustomer(uid:string, customer:Customer)  | 38        |
| getCustomer(uid:string) : Observable<Customer> | 38        |
| getUser(uid:string) : Observable<User>         | 38        |
| getCurrentUser()                               | 38        |
| getCurrentCustomer()                           | 39        |
| addEmployee(employee:Employee)                 | 39        |
| removeEmployee(employee:Employee)              | 39        |
| getEmployee(uid:string): Observable<Employee>  | 39        |
| getUser(uid:string) : Observable<User>         | 40        |

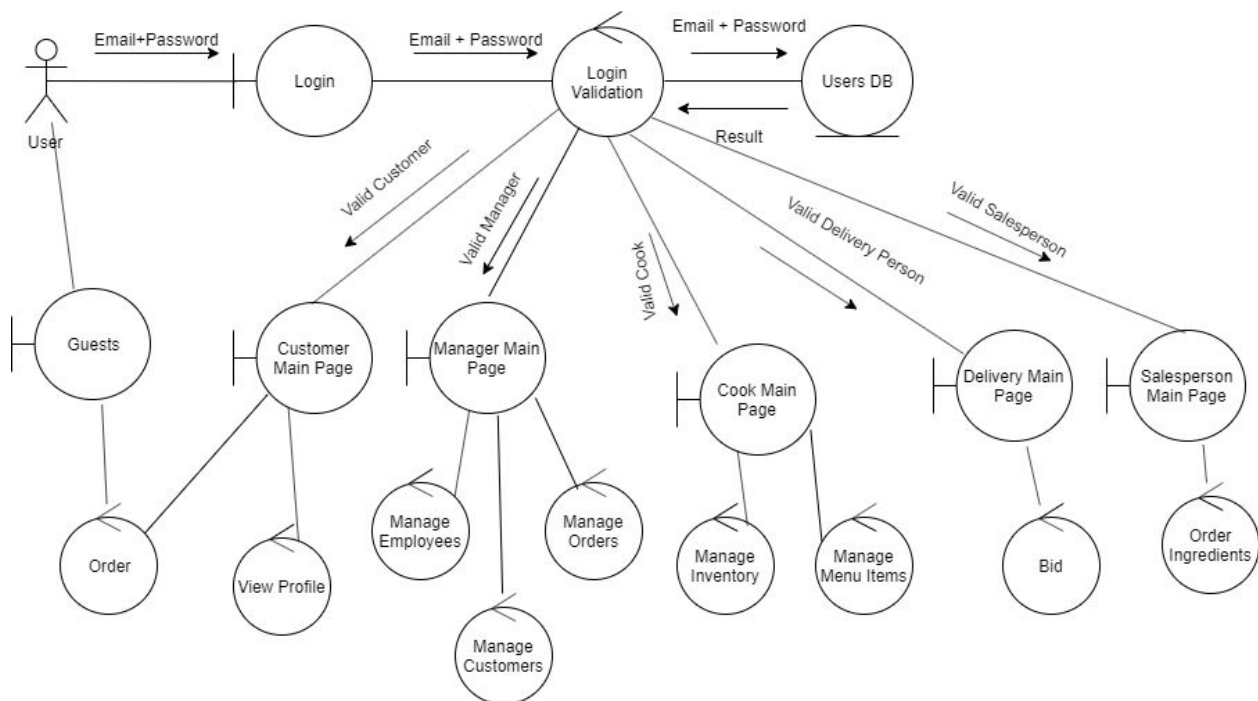
|  |           |
|--|-----------|
| addGuest(guest:Guest)                    | 40        |
| removeGuest(guest:Guest)                 | 40        |
| getGuest(uid:string) : Observable<Guest> | 41        |
| addOrder(order:Order)                    | 41        |
| removeOrder(order:Order)                 | 41        |
| updateOrder(uid:string, order:Order)     | 41        |
| getOrder(uid:string): Observable<Order>  | 42        |
| create(product:Product)                  | 42        |
| lookUp(uid:string): Observable<Product>  | 42        |
| update(uid:string, product:Product)      | 43        |
| delete(uid:string)                       | 43        |
| addUser(user:User)                       | 43        |
| removeUser(user:User)                    | 43        |
| getUser(uid:string): Observable<User>    | 44        |
| <b>System Screen</b>                     | <b>44</b> |
| <b>Minutes</b>                           | <b>47</b> |
| <b>Git Repo:</b>                         | <b>47</b> |

# 1.Introduction

Below you will encounter many diagrams, charts, and pseudocode, as they will assist in breaking down the design of our product. As you continue reading the paper, any questions or confusions that arise will be answered and clarified.

Not only does it allow customers (registered users) and guests (unregistered users) to order and receive food, but also gives managers, salespeople, cooks, and delivery people access to their own page to handle services in the company. Delivery people have access to see all the different orders from customers and guests, and bid on them. Salespeople are given comments from cooks, to know and order ingredients that are needed. Cooks are allowed to request more supplies from salespeople, rate salespeople, and change menu items. Managers can approve guest to customers, view order history, view all ratings, start delivery bidding process per order, pay employees, hire/fire employees, and remove warnings.

The collaboration class diagram below gives an overview on the entire Kitchen Zealot system.



DB: Database