DEPARTMENT OF
COMPUTER ENGINEERING

CENG 490

INITIAL DESIGN REPORT

Team Name : **QuartetSoft**

Project Name : **Sofr@m**

Team Members :

Evren Merih Tan - 1298314

Guner Mutlu - 1298017

Mustafa Zengin - 1250984

Rovsen Gasimov - 1222173

INDEX

1. Project Scope Definition

1.1 Problem Definition:

We are going to develop and implement a software restaurant automation system that has the following features; stock management, online reservation system, ordering via network enabled handheld computers, customer management, accounting, executive information system, statistical data about sales using graphics and customer profiles.

1.2 Project Scope:

After defining the general concept of the problem definition, it is time to describe the project scope. Our project is going to contain a database, an online reservation, and a bluetooth connection part that will connect the PDAs to the main computer. These three parts are concretized below :

1.2.1 *Scope of the Database Part:*

Database will hold data about customers, employees, online requests, stock, tables, PDAs, payments of the employees and statistics about sales.

Our database will have stock management support. Ingredients of a meal, how many units are needed for a meal, and quantity of meals that are sold will be hold in the database. The user will know ingredients that are left in his stock in order to control his stock in an efficient way.

In addition to stock management, he can create statistical data (daily, weekly, monthly, etc. sales, most favorite meal of customers, stock situation) and graphical representation of these data by using the related parts of the database model.

Furthermore, online reservations will be held in the database. The user will know the number of reserved tables and what type of meals are ordered by the customers who make an online reservation. Therefore orders of customers will be fulfilled in a rapid way when they come to the restaurant.

Another scope of the database part is that the user can hold customer profiles. So he can know what his customers' favourite meal are, special infos like birthday that will contribute to the pleasure of the customers.

Besides, employee information such as employee ID, date of employment, salary, employment type, etc. will be kept in the database. Concerning that, the user will know the general information about his employees.

On behalf of these, he can manage his account in accordance with the related data in database. For example, by calculating how much his daily expense and incomes, he can know his daily profit-loss statitistics.

1.2.2 *Scope of the Online Reservation Part:*

Our automation system is going to contain an online reservation part. This part will consist of a web-page and will be in touch with the database part. Customers can see the menu of the restaurant, price of the meals, the unoccupied tables without coming to the restaurant, in their homes, in their offices, etc. As online reservation is in touch with the database part, the user will know what customers' orders and compare them with his stock situation and can respond customers if their orders will be met or not. These attributes will ease the fulfillment of the customers' requests so that it is going to be easy to attract the customers' to the restaurant.

1.2.3 *Scope of the Bluetooth Part:*

Our system is going to have handheld computers which will be used by waiters to take the customers' orders. Waiters transfer these orders to the main computer as soon as they take the orders. Then, the main computer transmit these orders to the kitchen. If the ingredients are not sufficient to cook the meal, the kitchen will respond to the main computer and the main computer will transmit this message to the PDA and when the meal is ready, the kitchen will give a signal to the main computer and from there to the PDA so the waiter will understand that the meal is ready. These features are going to make the service in the restaurant faster and reduce the number of mistakes made by the waiters while taking the orders.

As a conclusion, our software is going to quicken the working mechanism in the restaurant. It is going to be easy to take statistical data and graphical representation of these data, to follow stock, reservation, employee situations and customer profiles, to reduce the service time, to increase the service quality.

2. CLASS DIAGRAMS

Here are the class diagrams used in initial design of our project. In the figure below, as it can be seen, Food class is the root abstract class. Other abstract classes namely EdibleMaterials and RawMaterials are derived from Food class. Drinks, Meals and Cereals, Vegetables leaf classes are derived from EdibleMaterials and RawMaterials classes respectevely. Here Meals class has a “has-a” relation with Cereals and Vegetables classes. In the similar manner OnlineRequest and RestaurantRequest classes are derived from Request abstract class. Again RestaurantRequest class has a Table and Request class has an Order class. Finally Order class has a “has-a” relation with Drinks and Meals classes.

In the second figure Person is the root abstract class. Waiter and Customer leaf classes are derived from this class. Waiter has Payment class and all Payment, Reservation Statistics and PDA classes have “has-a relation” with Date class. Reservation class has a Table class as well.

It is especially thought to design classes as modular as it can be. In the implementation phase of the project these abstract and root classes are going to be used widely. Carefully designed class diagrams will help a lot in that phase.

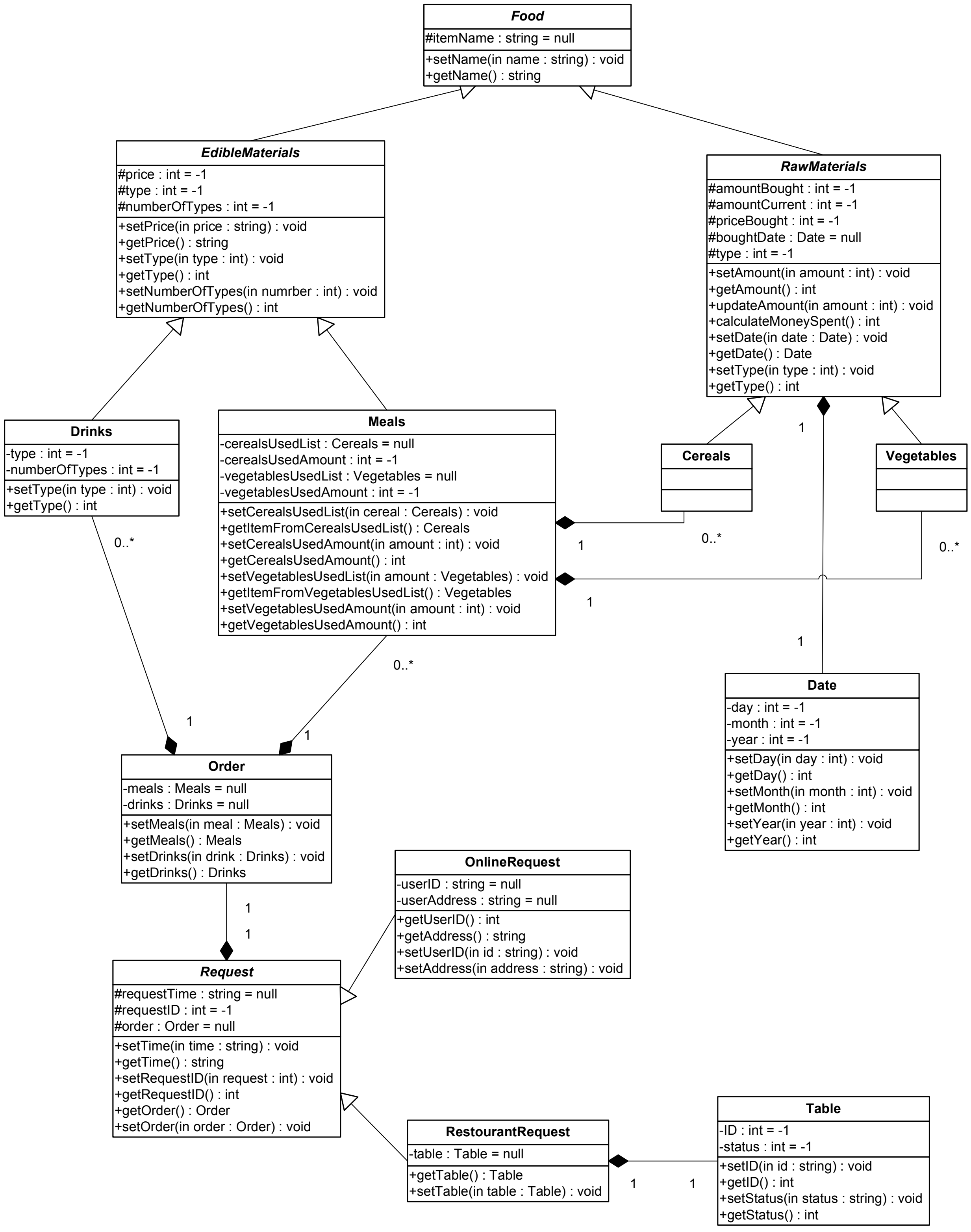

Food
#itemName : string = null
+setName(in name : string) : void
+getName() : string
EdibleMaterials
#price : int = -1
#type : int = -1
#numberOfTypes : int = -1
+setPrice(in price : string) : void
+getPrice() : string
+setType(in type : int) : void
+getType() : int
+setNumberOfTypes(in numrber : int) : void
+getNumberOfTypes() : int
RawMaterials
#amountBought : int = -1
#amountCurrent : int = -1
#priceBought : int = -1
#boughtDate : Date = null
#type : int = -1
+setAmount(in amount : int) : void
+getAmount() : int
+updateAmount(in amount : int) : void
+calculateMoneySpent() : int
+setDate(in date : Date) : void
+getDate() : Date
+setType(in type : int) : void
+getType() : int
Drinks
-type : int = -1
-numberOfTypes : int = -1
+setType(in type : int) : void
+getType() : int
Meals
-cerealsUsedList : Cereals = null
-cerealsUsedAmount : int = -1
-vegetablesUsedList : Vegetables = null
-vegetablesUsedAmount : int = -1
+setCerealsUsedList(in cereal : Cereals) : void
+getItemFromCerealsUsedList() : Cereals
+setCerealsUsedAmount(in amount : int) : void
+getCerealsUsedAmount() : int
+setVegetablesUsedList(in amount : Vegetables) : void
+getItemFromVegetablesUsedList() : Vegetables
+setVegetablesUsedAmount(in amount : int) : void
+getVegetablesUsedAmount() : int
Cereals
Vegetables
1
1
0..*
1
0..*
1
0..*
1
0..*
1
1
Date
-day : int = -1
-month : int = -1
-year : int = -1
+setDay(in day : int) : void
+getDay() : int
+setMonth(in month : int) : void
+getMonth() : int
+setYear(in year : int) : void
+getYear() : int
Order
-meals : Meals = null
-drinks : Drinks = null
+setMeals(in meal : Meals) : void
+getMeals() : Meals
+setDrinks(in drink : Drinks) : void
+getDrinks() : Drinks
OnlineRequest
-userID : string = null
-userAddress : string = null
+getUserID() : int
+getAddress() : string
+setUserID(in id : string) : void
+setAddress(in address : string) : void
1
1
Request
#requestTime : string = null
#requestID : int = -1
#order : Order = null
+setTime(in time : string) : void
+getTime() : string
+setRequestID(in request : int) : void
+getRequestID() : int
+getOrder() : Order
+setOrder(in order : Order) : void
RestourantRequest
-table : Table = null
+getTable() : Table
+setTable(in table : Table) : void
Table
-ID : int = -1
-status : int = -1
+setID(in id : string) : void
+getID() : int
+setStatus(in status : string) : void
+getStatus() : int
1
1

Figure 1

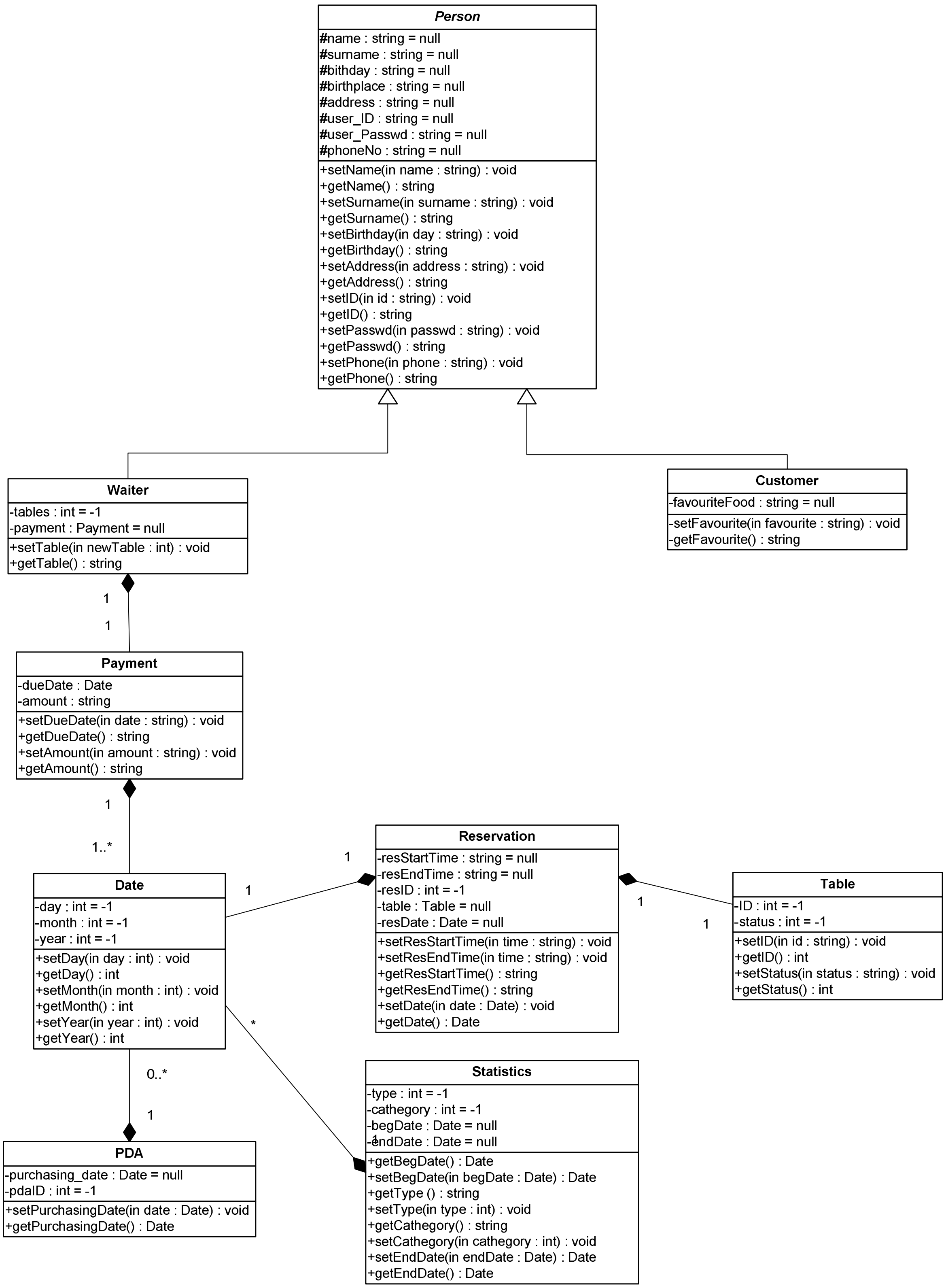
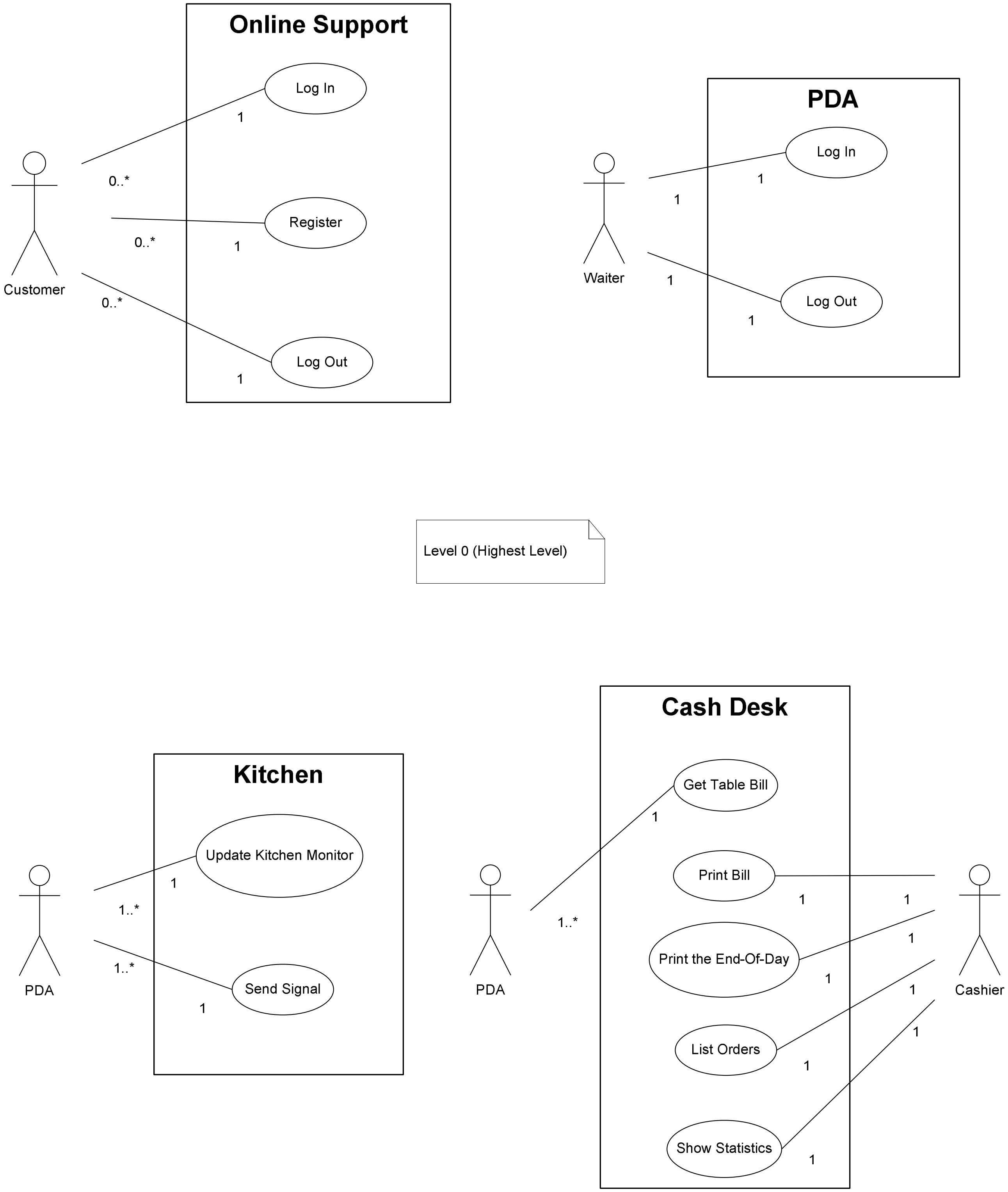

Person
#name : string = null
#surname : string = null
#bithday : string = null
#birthplace : string = null
#address : string = null
#user_ID : string = null
#user_Passwd : string = null
#phoneNo : string = null
+setName(in name : string) : void
+getName() : string
+setSurname(in surname : string) : void
+getSurname() : string
+setBirthday(in day : string) : void
+getBirthday() : string
+setAddress(in address : string) : void
+getAddress() : string
+setID(in id : string) : void
+getID() : string
+setPasswd(in passwd : string) : void
+getPasswd() : string
+setPhone(in phone : string) : void
+getPhone() : string
Waiter
-tables : int = -1
-payment : Payment = null
+setTable(in newTable : int) : void
+getTable() : string
Customer
-favouriteFood : string = null
-setFavourite(in favourite : string) : void
-getFavourite() : string
1
1
Payment
-dueDate : Date
-amount : string
+setDueDate(in date : string) : void
+getDueDate() : string
+setAmount(in amount : string) : void
+getAmount() : string
1
1..*
Reservation
-resStartTime : string = null
-resEndTime : string = null
-resID : int = -1
-table : Table = null
-resDate : Date = null
+setResStartTime(in time : string) : void
+setResEndTime(in time : string) : void
+getResStartTime() : string
+getResEndTime() : string
+setDate(in date : Date) : void
+getDate() : Date
1
1
1
1
Table
-ID : int = -1
-status : int = -1
+setID(in id : string) : void
+getID() : int
+setStatus(in status : string) : void
+getStatus() : int
Date
-day : int = -1
-month : int = -1
-year : int = -1
+setDay(in day : int) : void
+getDay() : int
+setMonth(in month : int) : void
+getMonth() : int
+setYear(in year : int) : void
+getYear() : int
*
0..*
1
Statistics
-type : int = -1
-cathegory : int = -1
-begDate : Date = null
-endDate : Date = null
1
+getBegDate() : Date
+setBegDate(in begDate : Date) : Date
+getType () : string
+setType(in type : int) : void
+getCathegory() : string
+setCathegory(in cathegory : int) : void
+setEndDate(in endDate : Date) : Date
+getEndDate() : Date
PDA
-purchasing_date : Date = null
-pdaID : int = -1
+setPurchasingDate(in date : Date) : void
+getPurchasingDate() : Date

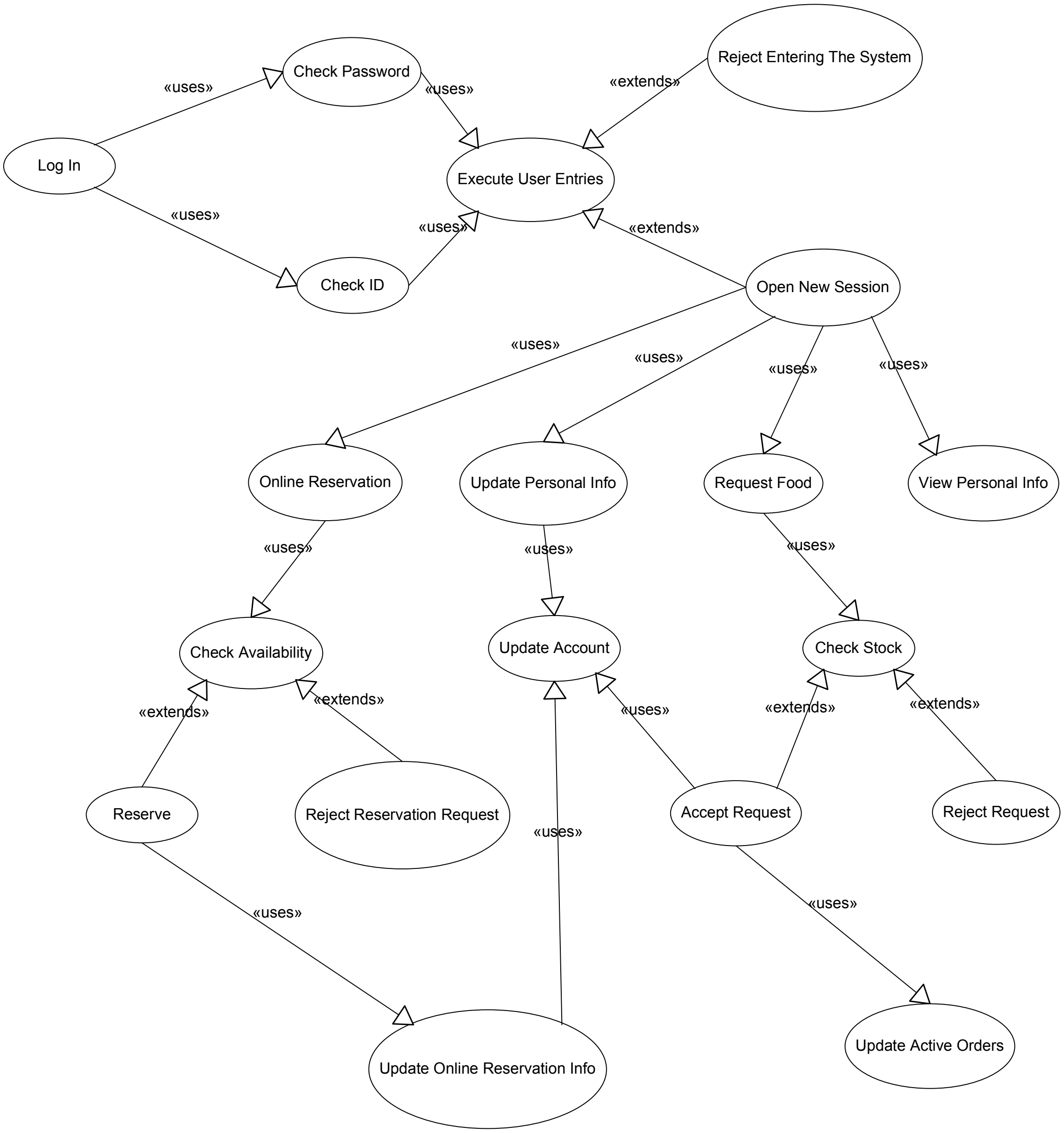
Figure 2

3. USE - CASE DIAGRAMS

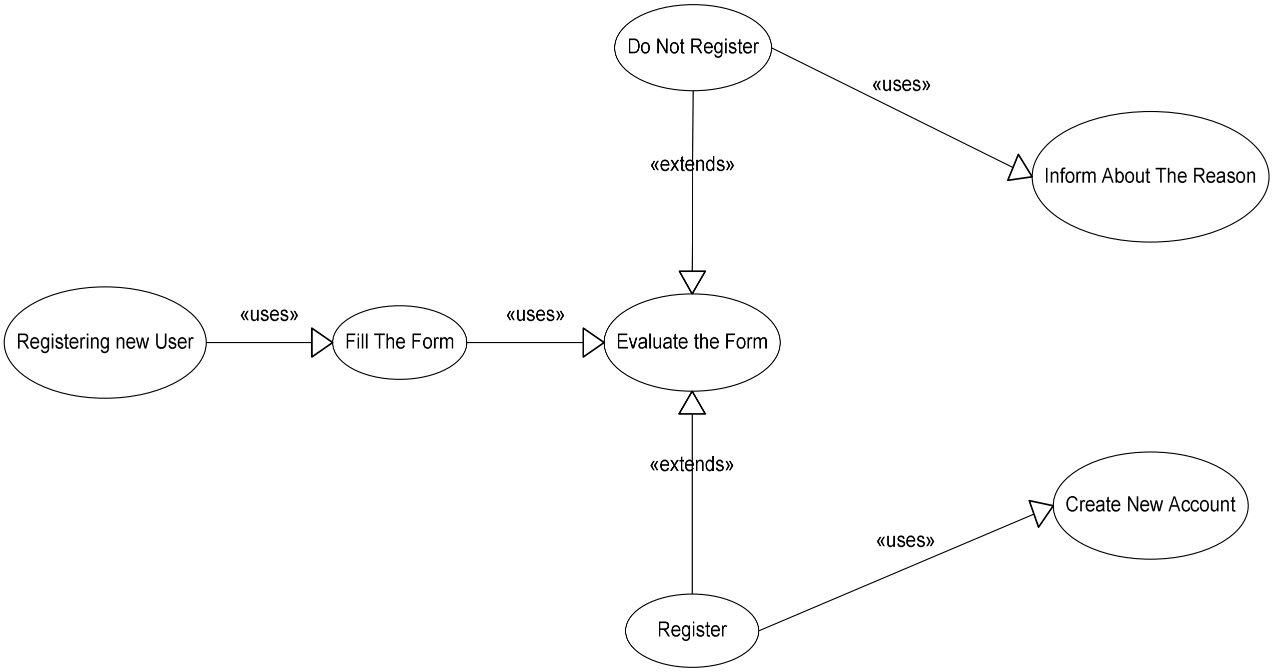
In the figure below Use-Case diagrams represented at the highest level. As can be seen we have 4 parts mainly. Large rectangles represent system boundary, where out of it there is an actor. Customer, waiter, PDA and cashier are actors, where Online Support, PDA, Kitchen and Cash Desk are the system boundaries.

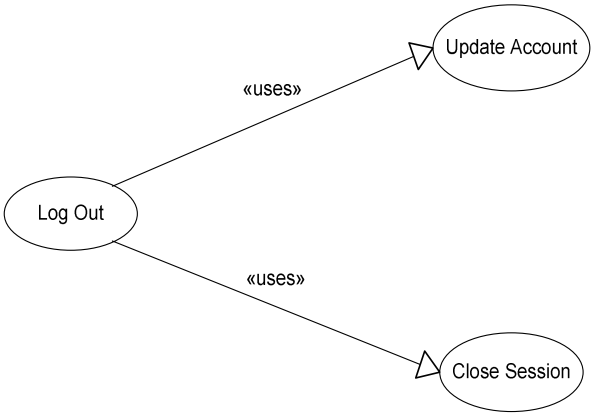


Use – Case 1 (Customer - Online Support): For this Use – Case login action is described as in below figure. Here Customer has to login first, to take an action like Online Reservation, Update Personal Info, View Personal Info or Request Food. Each change in stock or in account results with the update of stock or personal account.

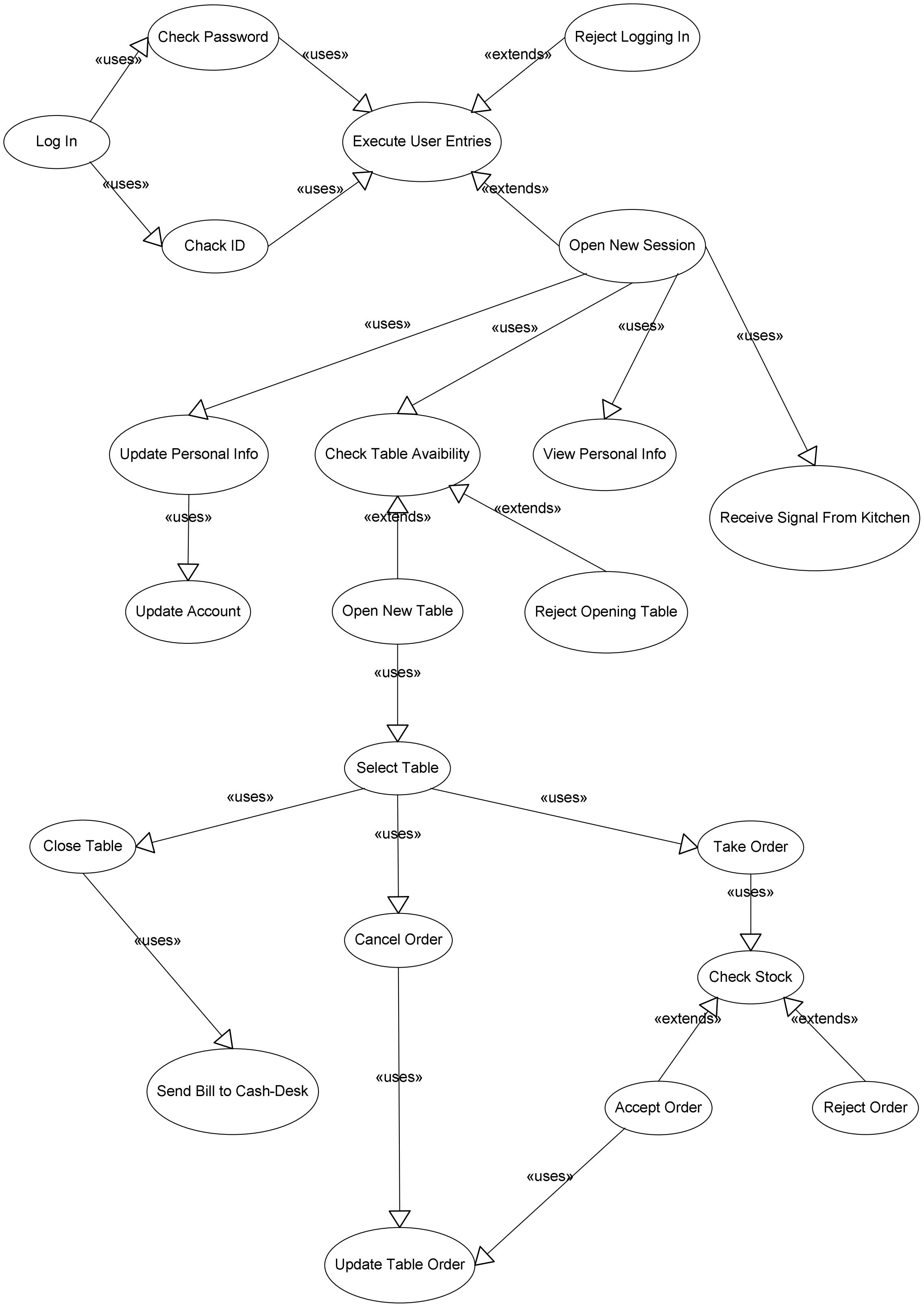


The figures below show registration and log in actions respectevely. Log out action results in update of account.

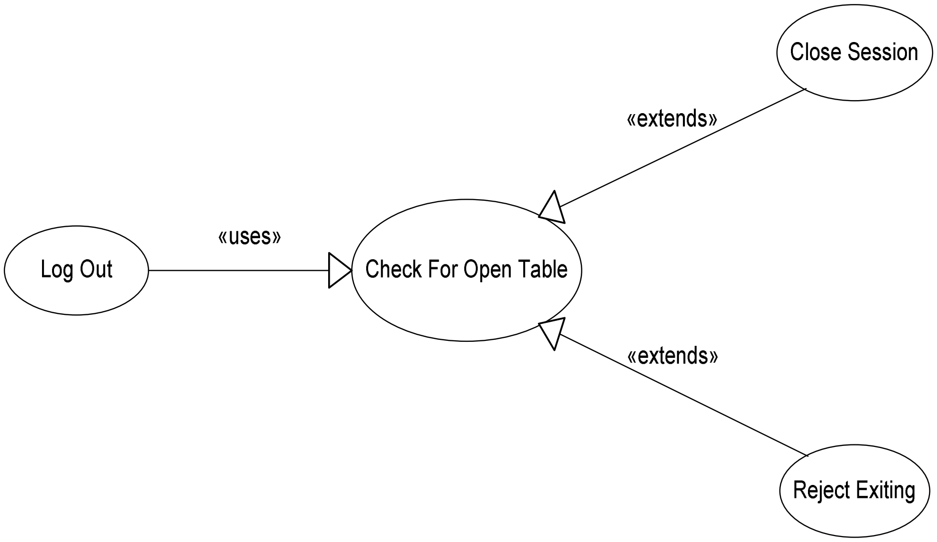


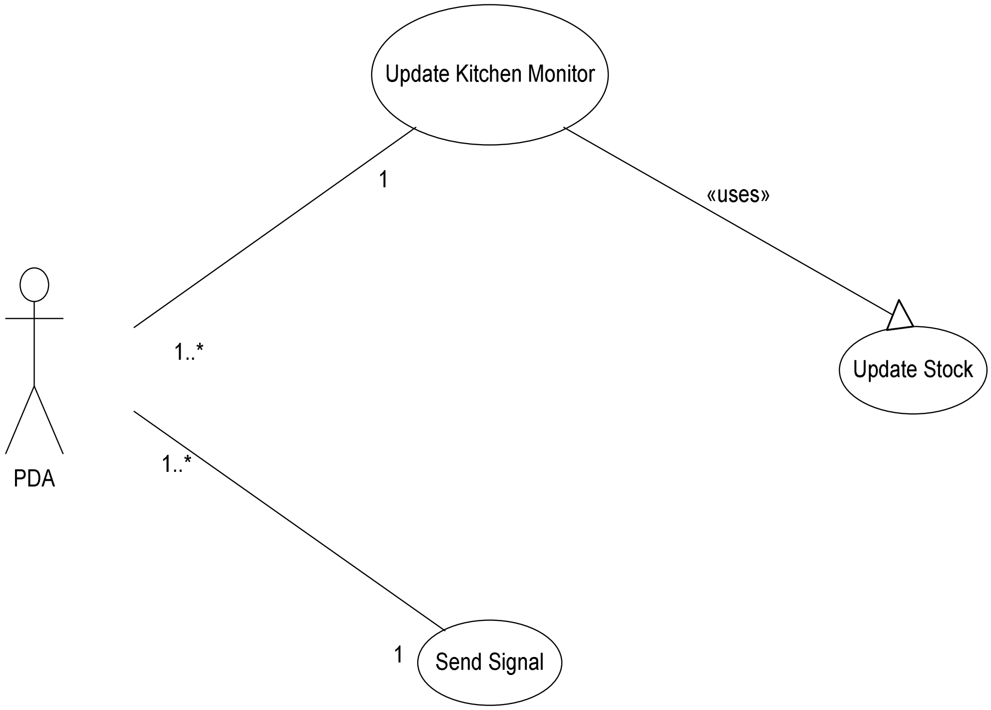


Use – Case 2 (Waiter - PDA): Below action is the log in part of the second Use – Case diagram. Here again to take an action waiter has to log in. Here again Update Personal Info, Check Table Availibility, View Personal Info and Receive Signal from Kitchen are main actions to take.

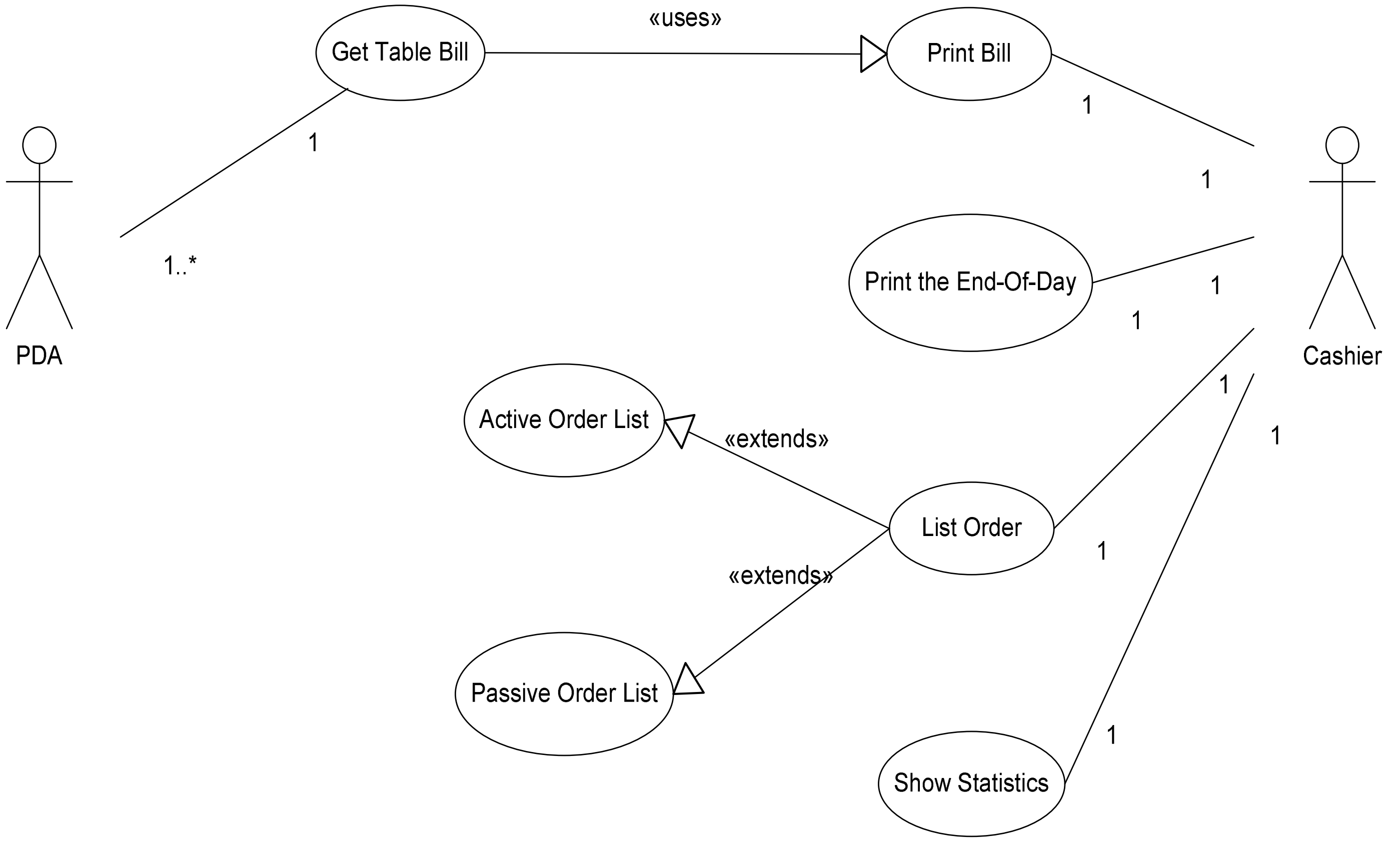


Log out and update kitchen, send signal parts are given below. When meal is ready, some signal is generated in the kitchen and send to the PDA.





Use – Case 3 (PDA – Cash Desk – Cashier): In this Use - Case there are 2 main actors, PDA and Cashier namely. The only action that PDA participates in is Get Table Bill. Cashier's actions are Print Bill, Print the End-Of-Day, List Order and Show Statistics.



4. COLLABORATION DIAGRAMS

Take Order(Waiter)

Waiter logs in to PDA and takes order if it is available, after this the order added to *Active Orderlist*

Cancel Order(Waiter)

Waiter logs in to PDA and sends cancel the order signal, after this the canceled order removed from *Active OrderList.*

View Personel Info(Waiter)

Waiter logs in to PDA and requests view info,after this *Personel Account System* shows the info.

Update Personel Info(Waiter)

Waiter logs in to PDA and requests update info,after this Personel Account System loads his profile,he makes changes in his profile and requests save,the request replied by Personel Account System as updating the profile.

Open && Send Bill of Table(Waiter)

Waiter logs in to PDA and requests open new table, the table is selected from *tables* and marked as “busy” .If another table need to be served ,waiter can select it from *tables* and can service.Besides these, if bill request came from costumers, he sends the bill to *cash-desk*.

Process in Kitchen(PDA)

When order comes to kitchen, it is displayed on the kitchen monitor.Cook prepares the order and sends order is ready signal.

Registration(Customer)

Customer enters to webservice and requests sign up, Personel Account System replies to request and creates new account.

Online Reservation(Customer)

Customer logs in, selects his/her table from list and sends reserve signal,the request is replied from *tables* and it returnes a result to customer.If the result is positive, table is assigned to customer and changes its status to reserved for a time interval.

Update Personel Info(Customer)

Customer logs in, requests change info, Personel Account System saves the changes.

View Personel Info(Customer)

Customer logs in, requests view info, Personel Account System replies to request and shows the info.

Request Food(Customer)

Custmer logs in, requests food(at this time menu is listed and user selects from list).His/her request puts in a queue and checked in the stock.If it is available, a positive reply is given and enqueued to *Active Order List.*

Create Bill(PDA)

PDA sends bill request of a table with an ID,the request puts in a queue *(BillList).Bill List* sends bills to printer via cash-desk in a loop.

Get End Of Day(Cash-Desk)

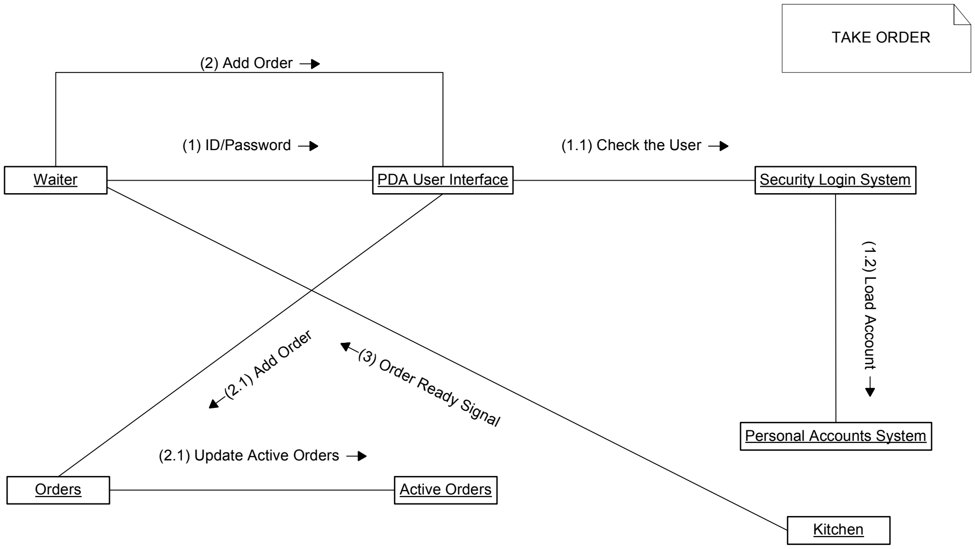
Cash-Desk logs in and requests get end of day,the request is handled by *Passive Orders List.* It sends bills to executive system.It executes and produces a result.The result is sent to printer.

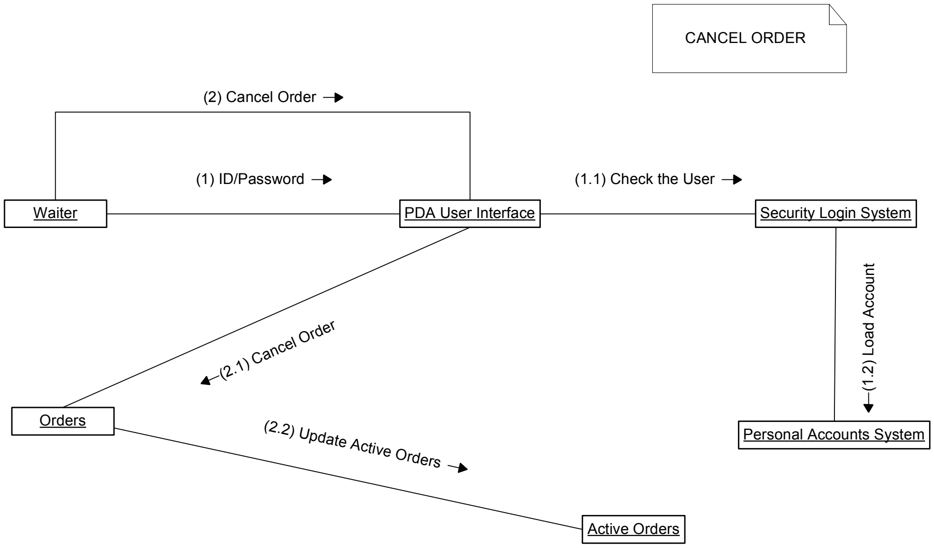
List Orders(Cash-Desk)

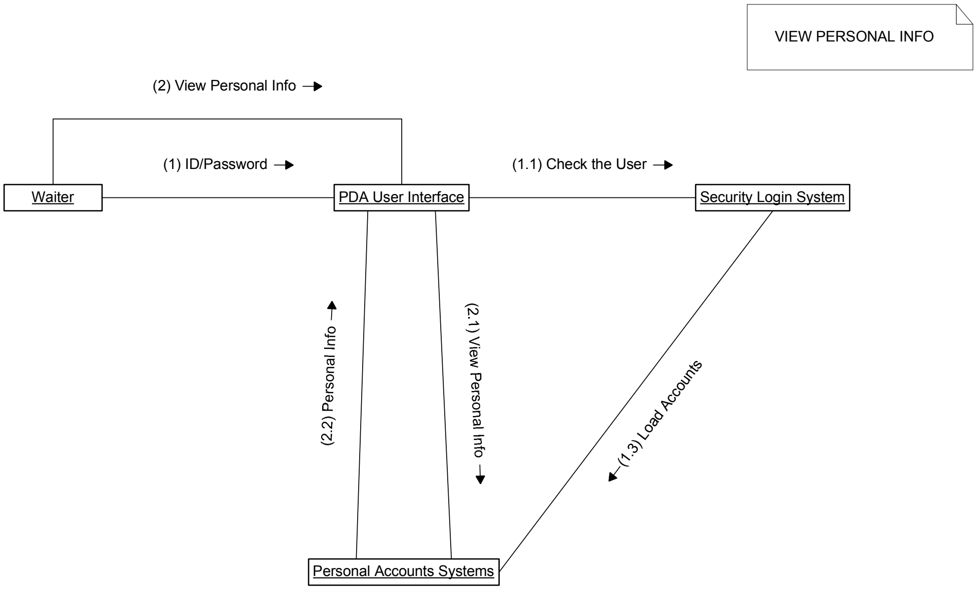
Cash-Desk logs in and requests list orders,the request is handled by *Active Orders List.*It shows orders on the monitor.

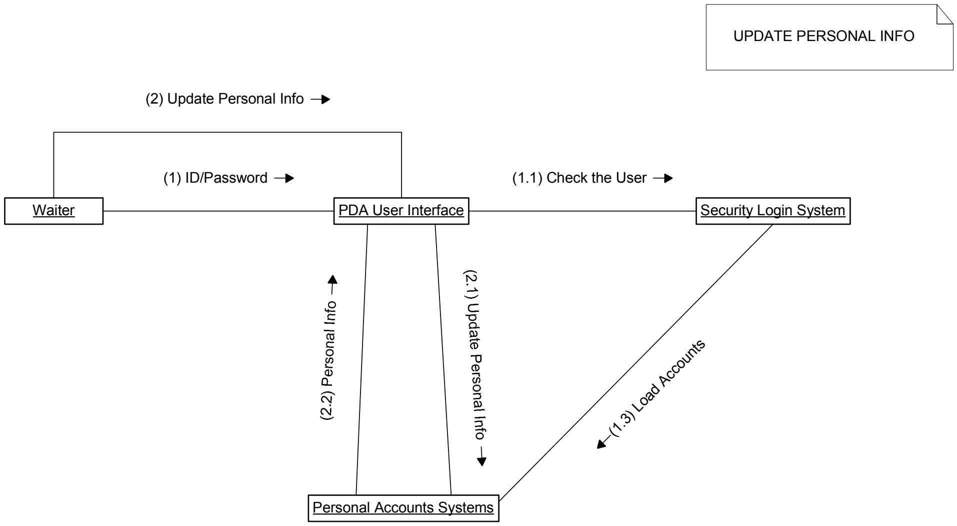
Generate Statistics(Cash-Desk)

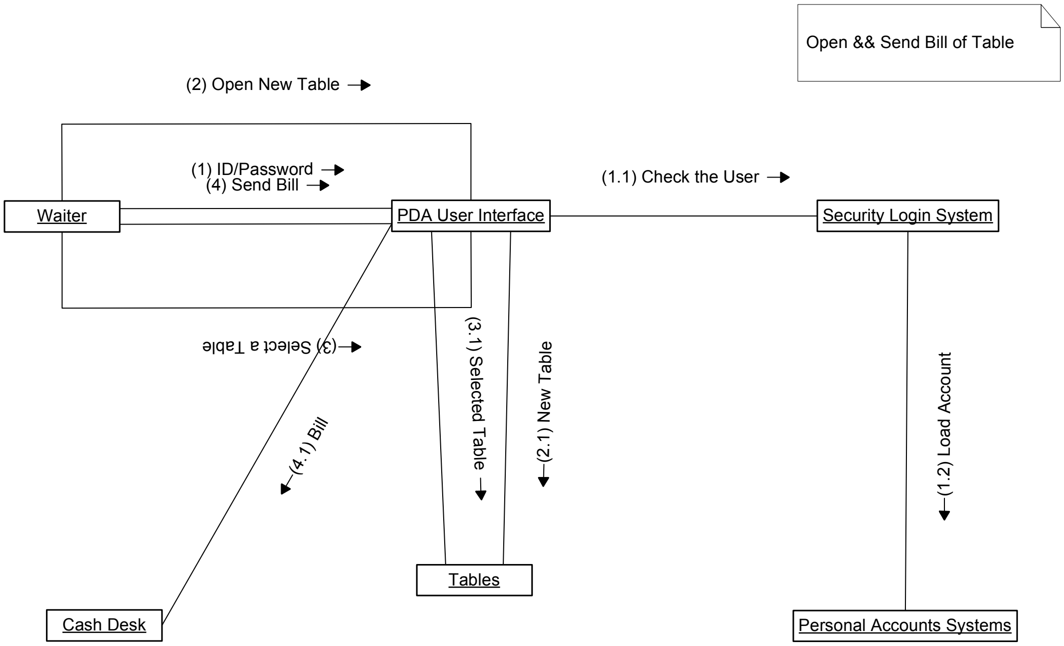
Cash-Desk logs in and requests generate statistics, the request is handled by statistics class.It takes the orders from *Passive Order List*, generate statistics and show them on the monitor.

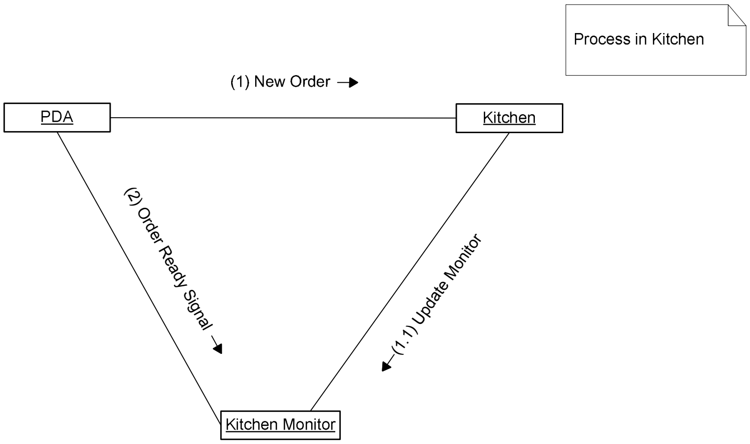

TAKE ORDER
(2) Add Order →
(1) ID/Password →
(1.1) Check the User →
Waiter
PDA User Interface
Security Login System
(1.2) Load Account →
Personal Accounts System
← (2.1) Add Order
← (3) Order Ready Signal
(2.1) Update Active Orders →
Orders
Active Orders
Kitchen

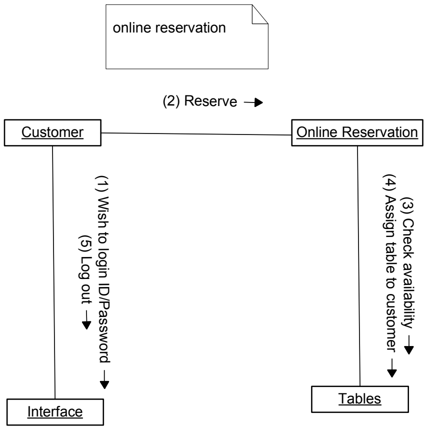

CANCEL ORDER
(2) Cancel Order →
(1) ID/Password →
(1.1) Check the User →
Waiter
PDA User Interface
Security Login System
(1.2) Load Account →
← (2.1) Cancel Order
Orders
Personal Accounts System
(2.2) Update Active Orders →
Active Orders

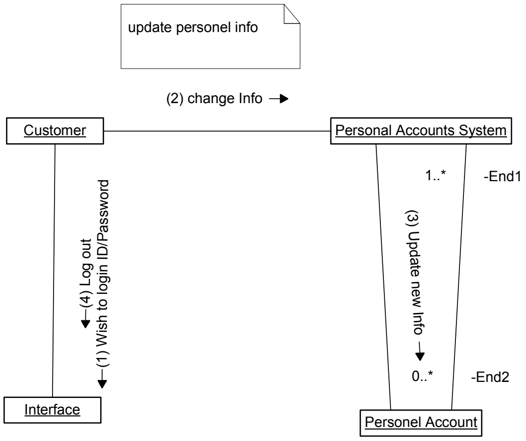

VIEW PERSONAL INFO
(2) View Personal Info →
(1) ID/Password →
(1.1) Check the User →
Waiter
PDA User Interface
Security Login System
(2.2) Personal Info →
(2.1) View Personal Info →
←(1.3) Load Accounts
Personal Accounts Systems

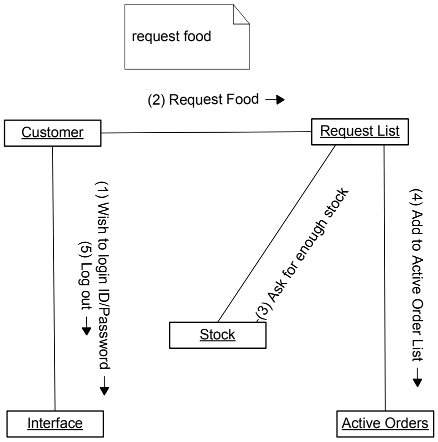

UPDATE PERSONAL INFO
(2) Update Personal Info →
(1) ID/Password →
(1.1) Check the User →
Waiter
PDA User Interface
Security Login System
(2.2) Personal Info →
(2.1) Update Personal Info →
←(1.3) Load Accounts
Personal Accounts Systems

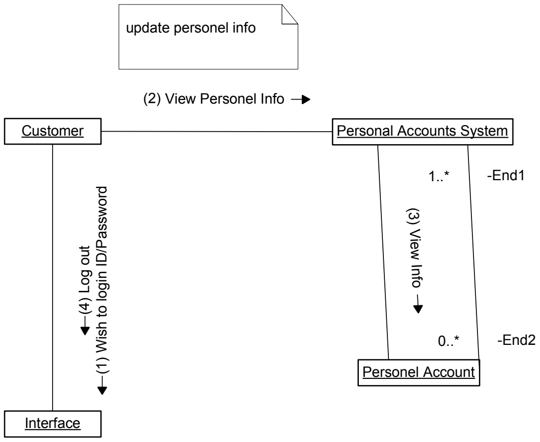

Open && Send Bill of Table
(2) Open New Table
(1) ID/Password
(4) Send Bill
(1.1) Check the User
Waiter
PDA User Interface
Security Login System
(3) Select a Table
(3.1) Selected Table
(2.1) New Table
(4.1) Bill
(1.2) Load Account
Tables
Cash Desk
Personal Accounts Systems

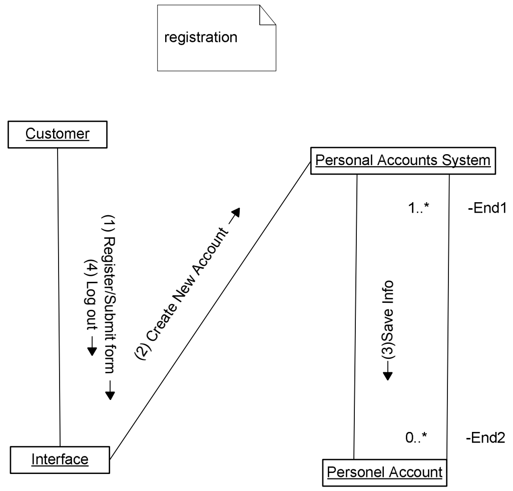

Process in Kitchen
(1) New Order
PDA
Kitchen
(2) Order Ready Signal
(1.1) Update Monitor
Kitchen Monitor

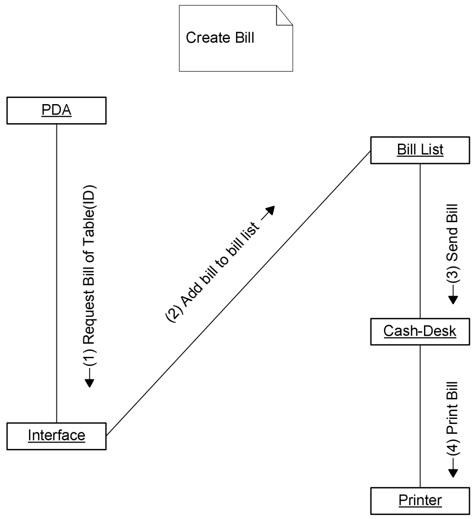

online reservation
(2) Reserve →
Customer
Online Reservation
(1) Wish to login ID/Password →
(5) Log out →
(3) Check availability →
(4) Assign table to customer →
Interface
Tables

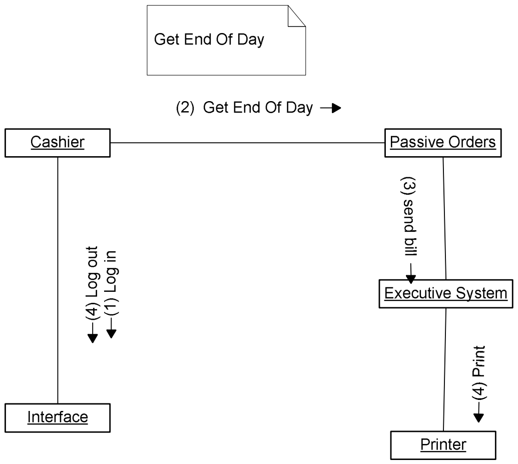

update personel info
(2) change Info →
Customer
Personal Accounts System
1..*
-End1
(4) Log out
(1) Wish to login ID/Password
(3) Update new Info →
0..*
-End2
Interface
Personel Account

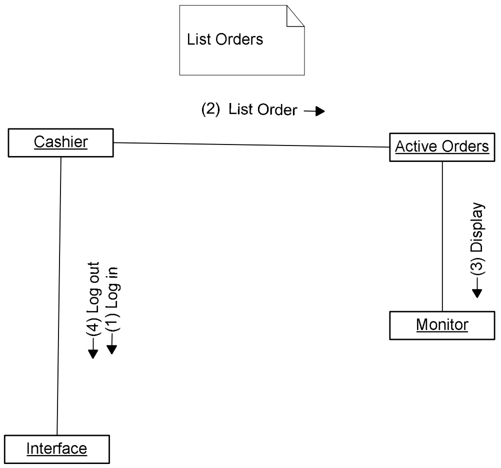

request food
(2) Request Food →
Customer
Request List
(1) Wish to login ID/Password →
(5) Log out →
(3) Ask for enough stock
(4) Add to Active Order List →
Stock
Interface
Active Orders

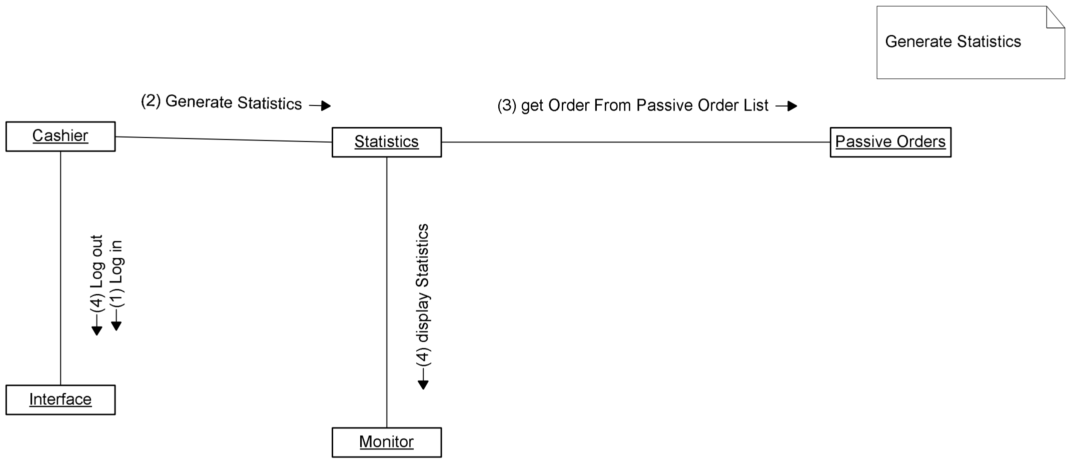

update personel info
(2) View Personel Info →
Customer
Personal Accounts System
1..*
-End1
(4) Log out
(1) Wish to login ID/Password
(3) View Info →
0..*
-End2
Personel Account
Interface


registration
Customer
Interface
Personal Accounts System
Personel Account
(1) Register/Submit form
(4) Log out
(2) Create New Account
(3)Save Info
1..*
-End1
0..*
-End2

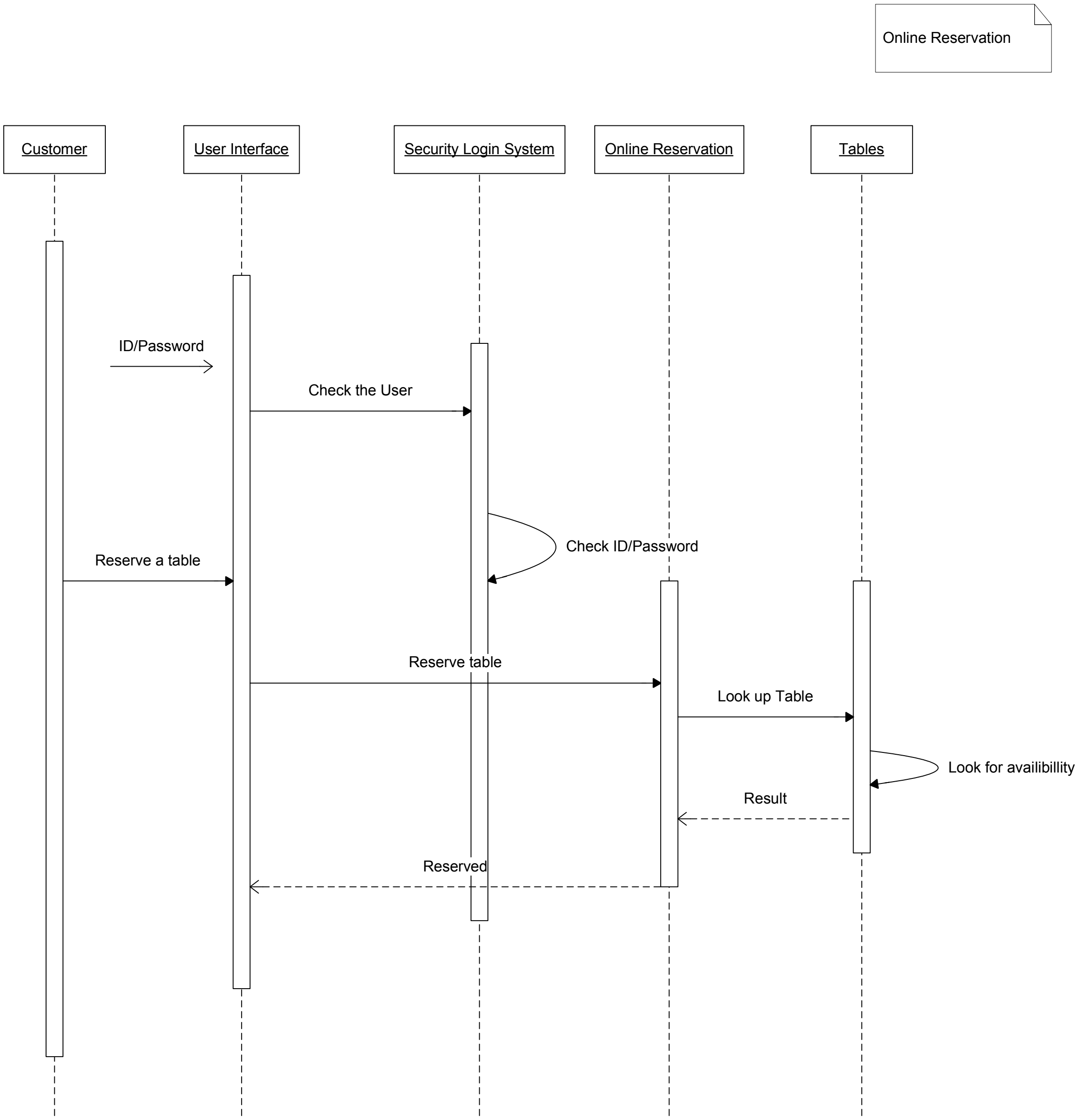

Create Bill
PDA
Interface
Bill List
Cash-Desk
Printer
(1) Request Bill of Table(ID)
(2) Add bill to bill list
(3) Send Bill
(4) Print Bill

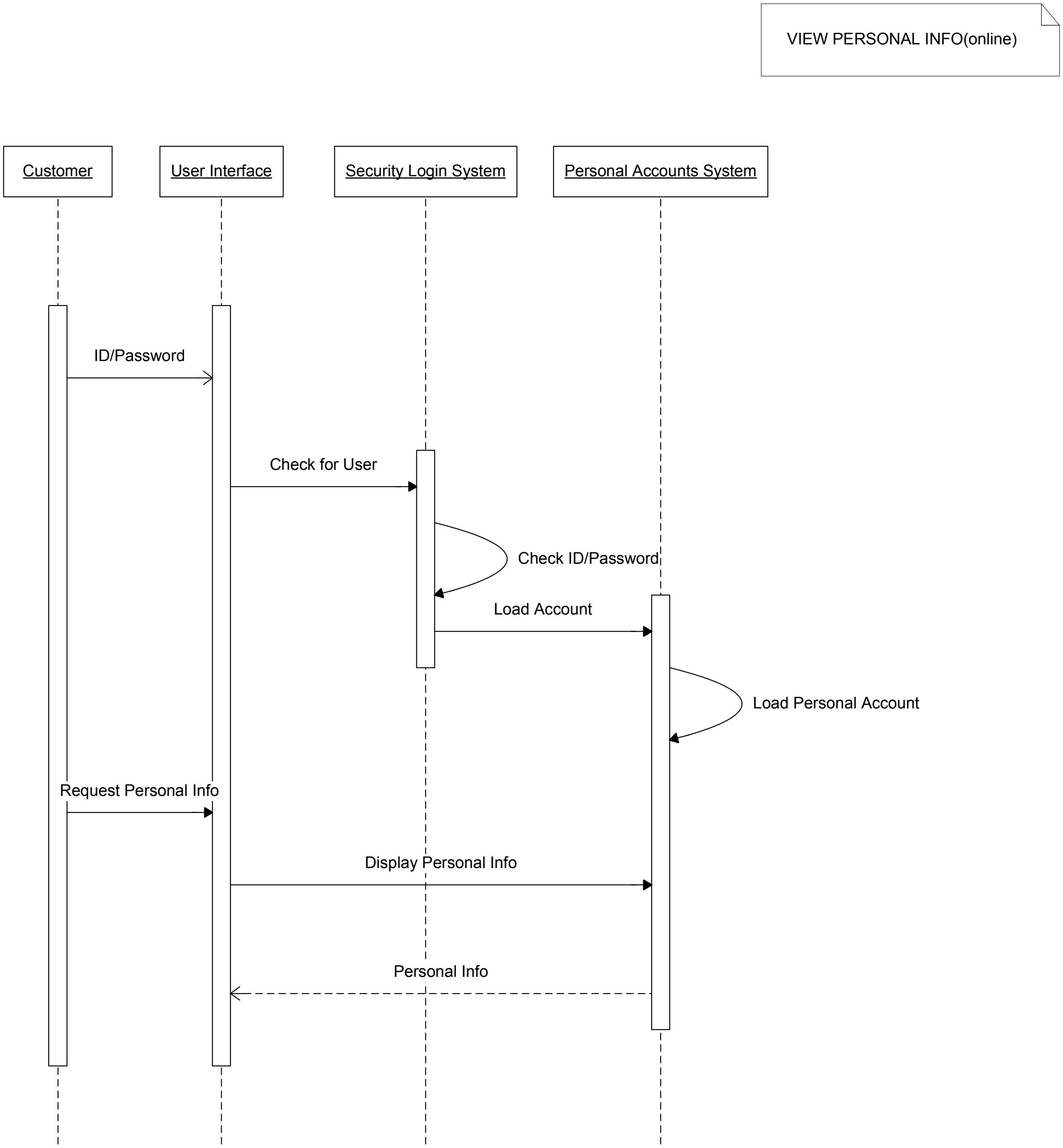

Get End Of Day
(2) Get End Of Day
Cashier
Passive Orders
Executive System
Interface
Printer
(3) send bill
(4) Log out
(1) Log in
(4) Print

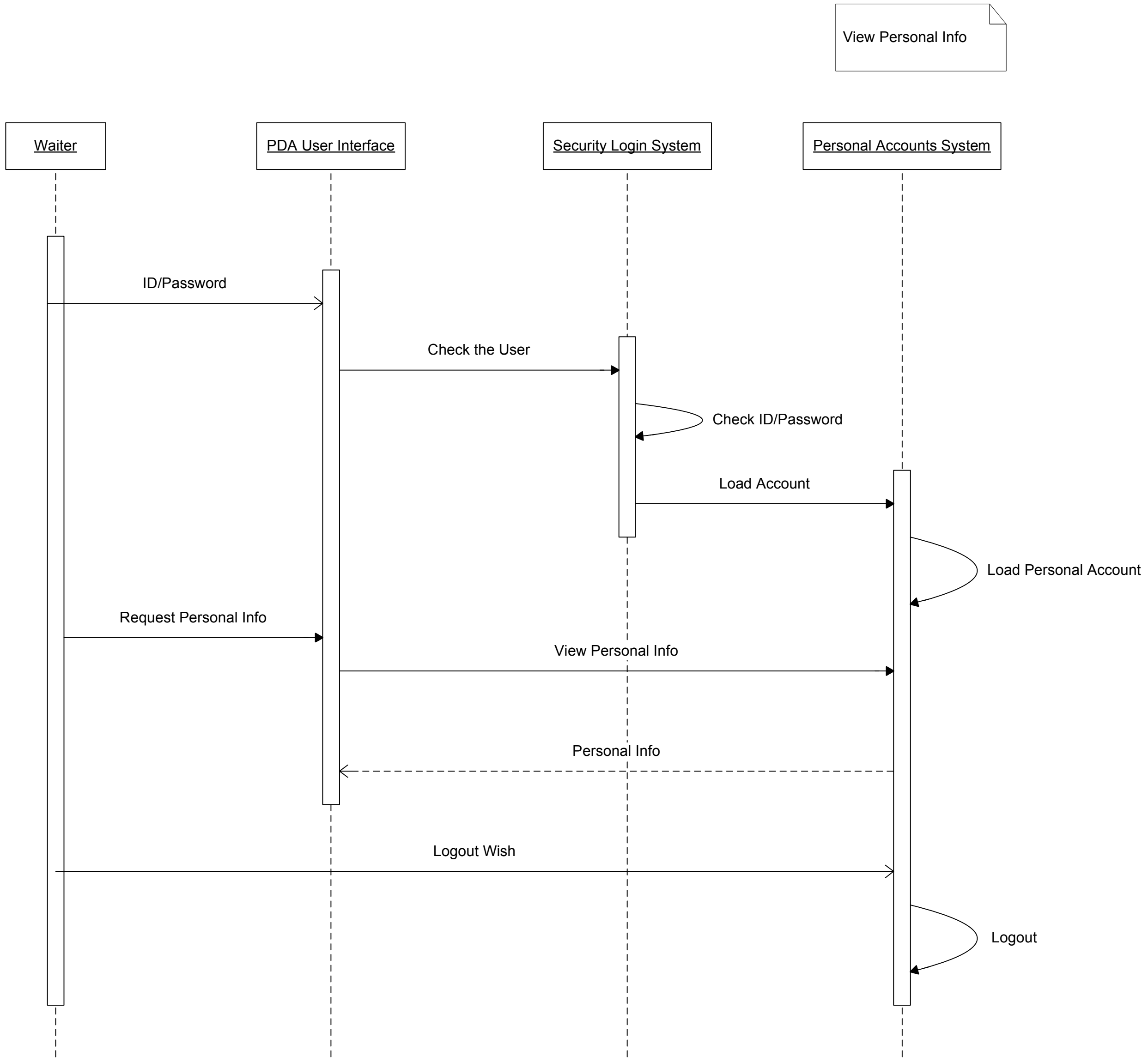

List Orders
(2) List Order
Cashier
Active Orders
Monitor
Interface
(4) Log out
(1) Log in
(3) Display

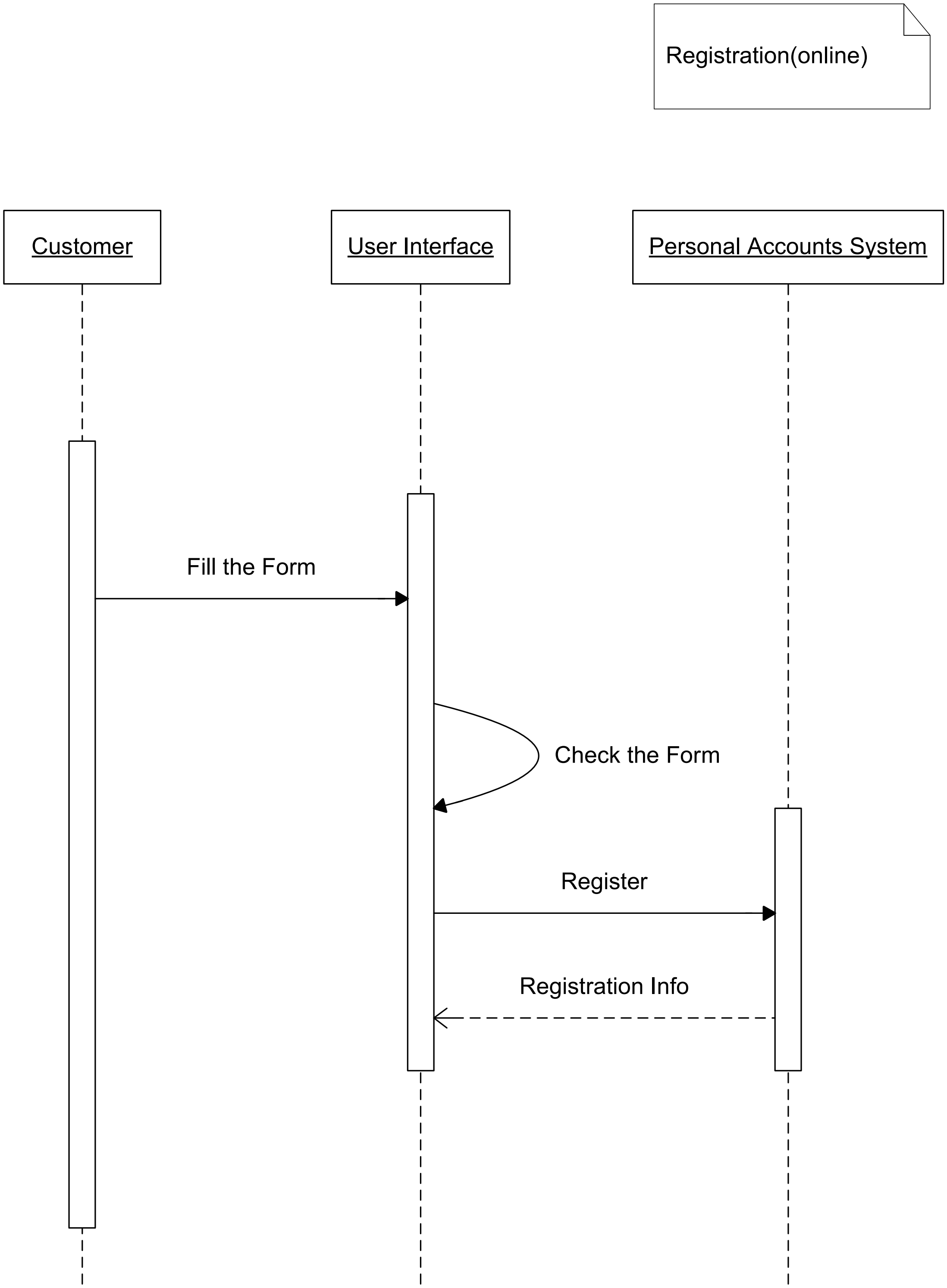

Generate Statistics
(2) Generate Statistics
(3) get Order From Passive Order List
Cashier
Statistics
Passive Orders
(4) Log out
(1) Log in
(4) display Statistics
Interface
Monitor

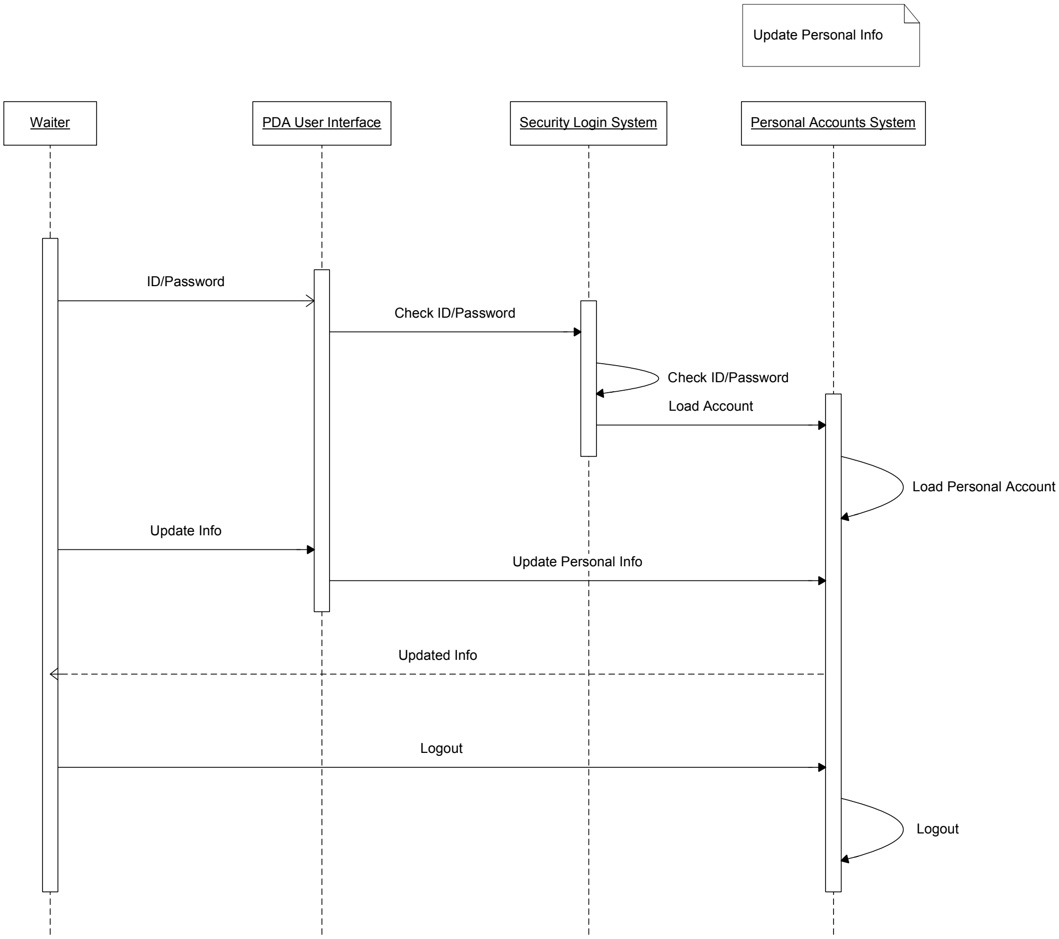
5. SEQUENCE DIAGRAMS

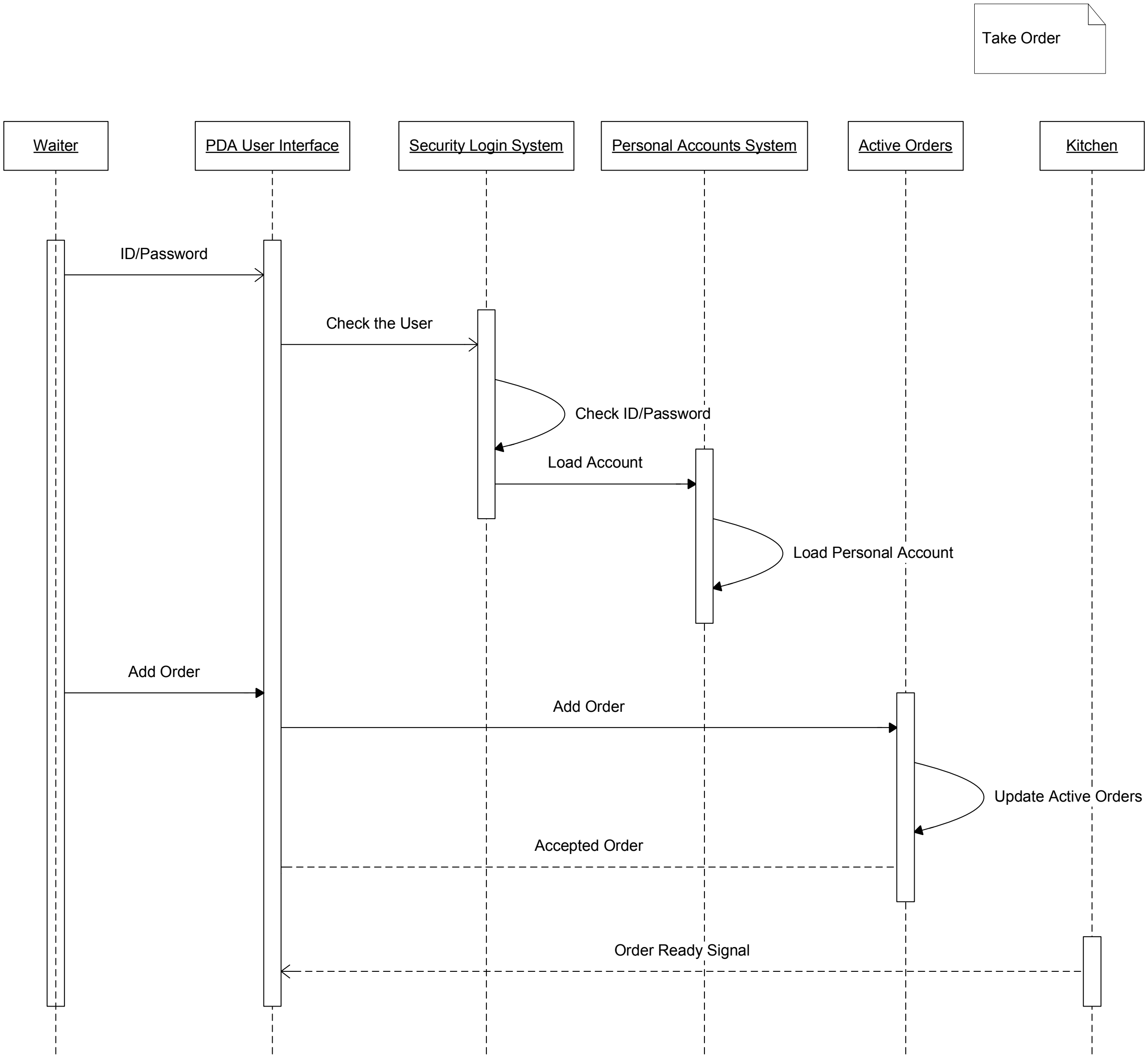


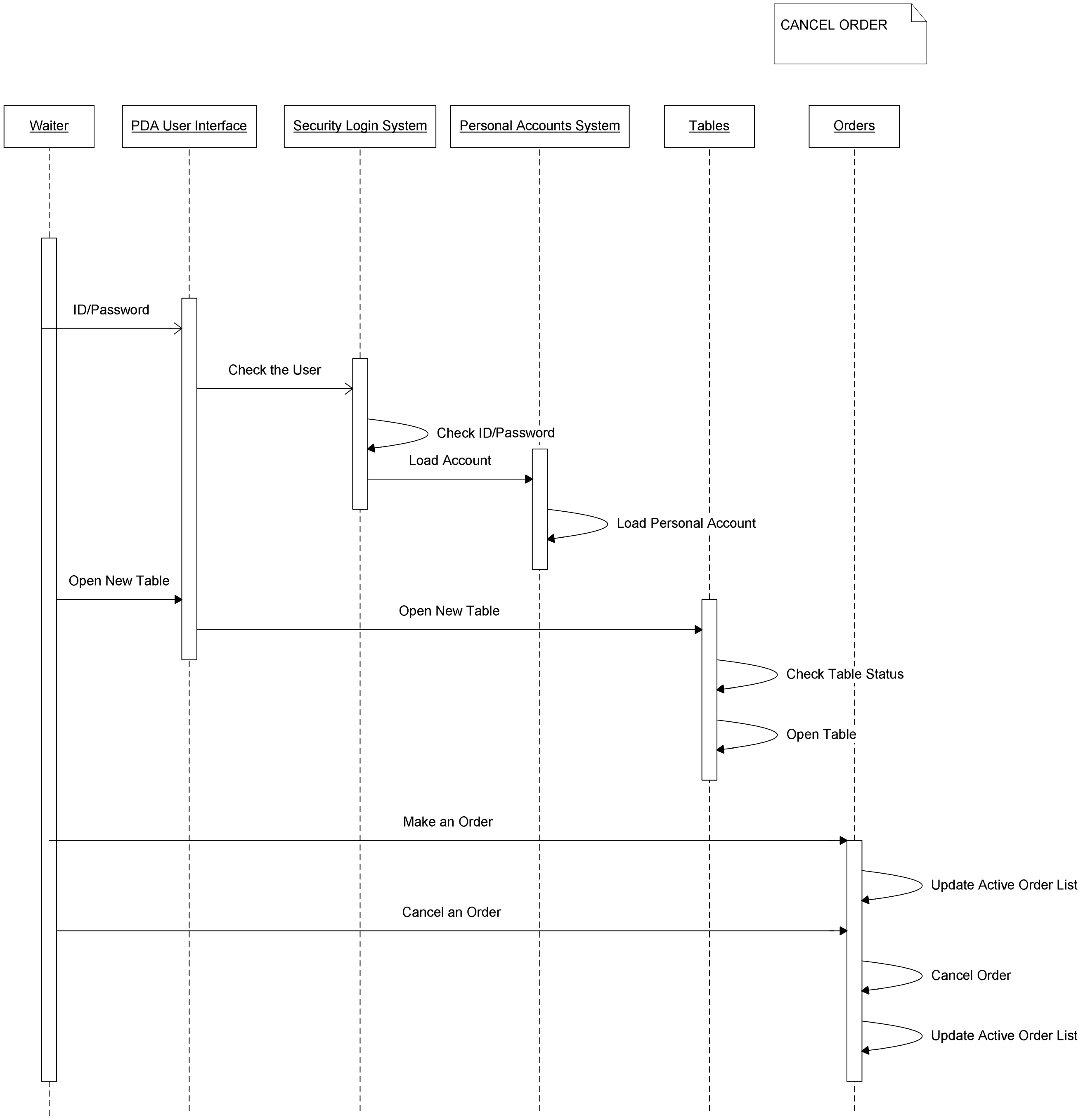

VIEW PERSONAL INFO(online)
Customer
User Interface
Security Login System
Personal Accounts System
ID/Password
Check for User
Check ID/Password
Load Account
Load Personal Account
Request Personal Info
Display Personal Info
Personal Info

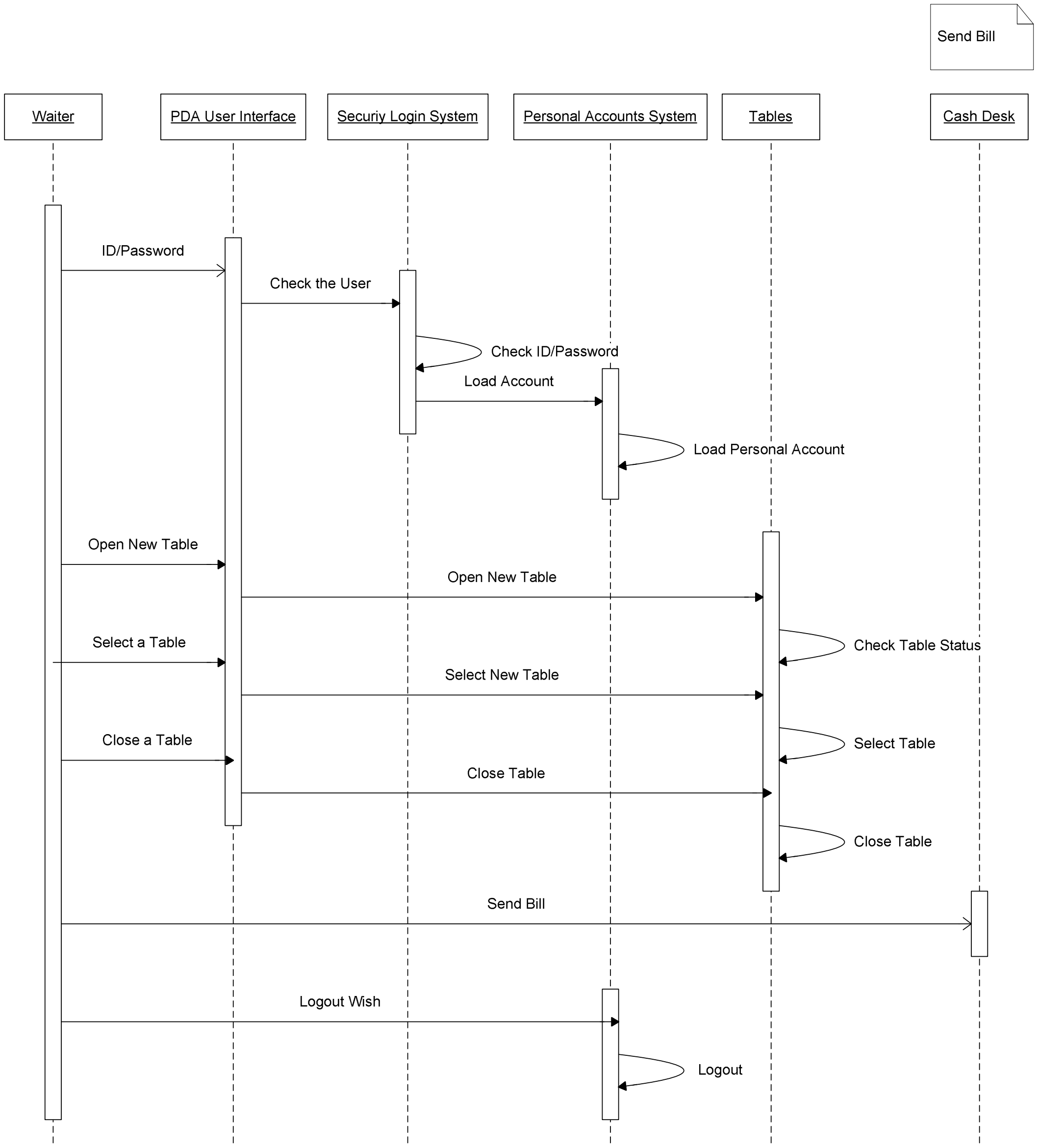

View Personal Info
Waiter
PDA User Interface
Security Login System
Personal Accounts System
ID/Password
Check the User
Check ID/Password
Load Account
Load Personal Account
Request Personal Info
View Personal Info
Personal Info
Logout Wish
Logout

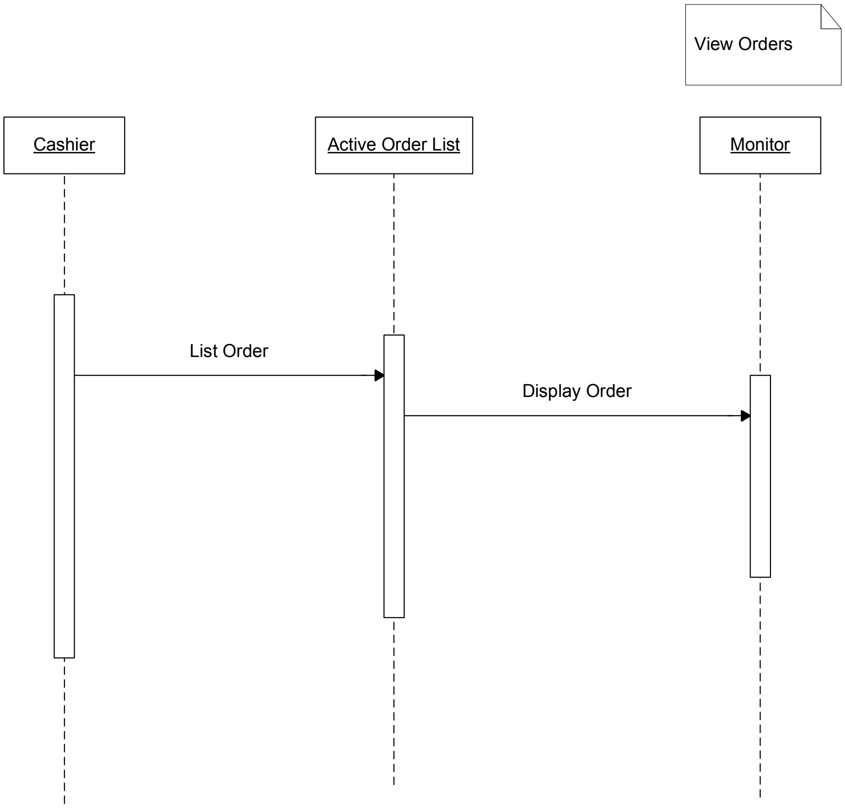

Registration(online)
Customer
User Interface
Personal Accounts System
Fill the Form
Check the Form
Register
Registration Info

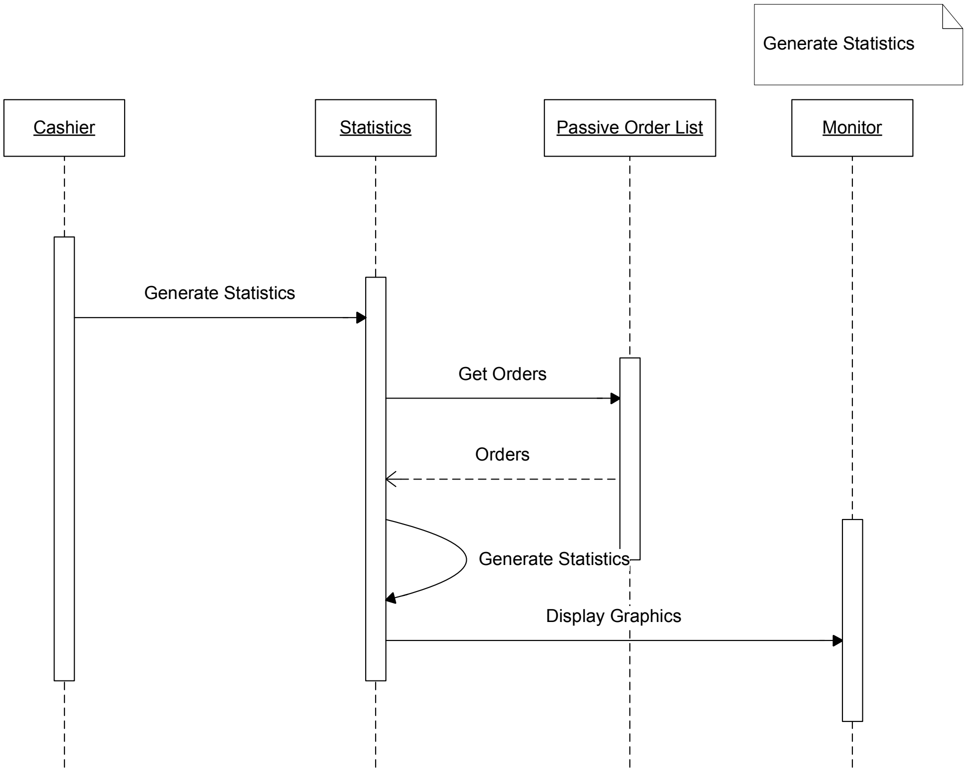

Update Personal Info
Waiter
PDA User Interface
Security Login System
Personal Accounts System
ID/Password
Check ID/Password
Check ID/Password
Load Account
Load Personal Account
Update Info
Update Personal Info
Updated Info
Logout
Logout

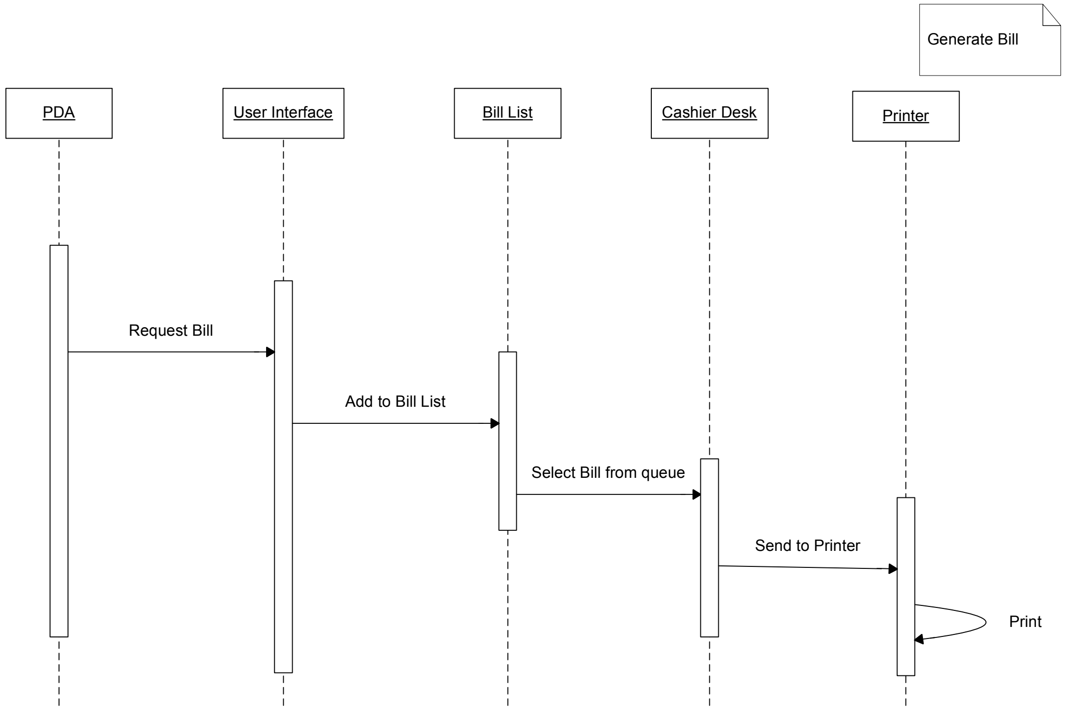

Take Order
Waiter
PDA User Interface
Security Login System
Personal Accounts System
Active Orders
Kitchen
ID/Password
Check the User
Check ID/Password
Load Account
Load Personal Account
Add Order
Add Order
Update Active Orders
Accepted Order
Order Ready Signal

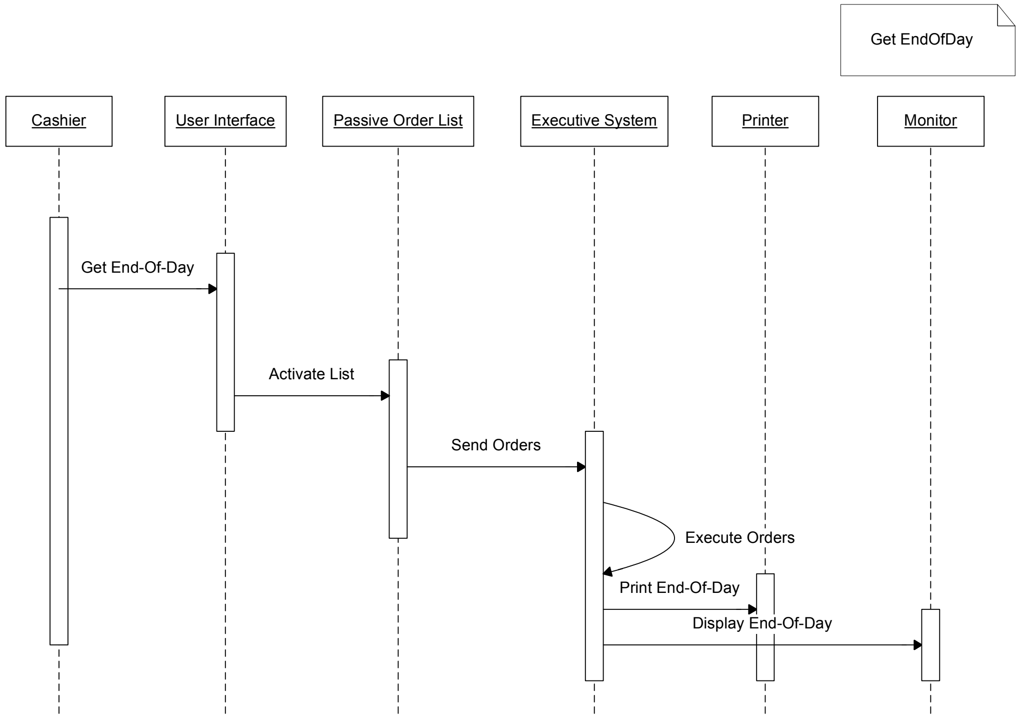

CANCEL ORDER
Waiter
PDA User Interface
Security Login System
Personal Accounts System
Tables
Orders
ID/Password
Check the User
Check ID/Password
Load Account
Load Personal Account
Open New Table
Open New Table
Check Table Status
Open Table
Make an Order
Update Active Order List
Cancel an Order
Cancel Order
Update Active Order List

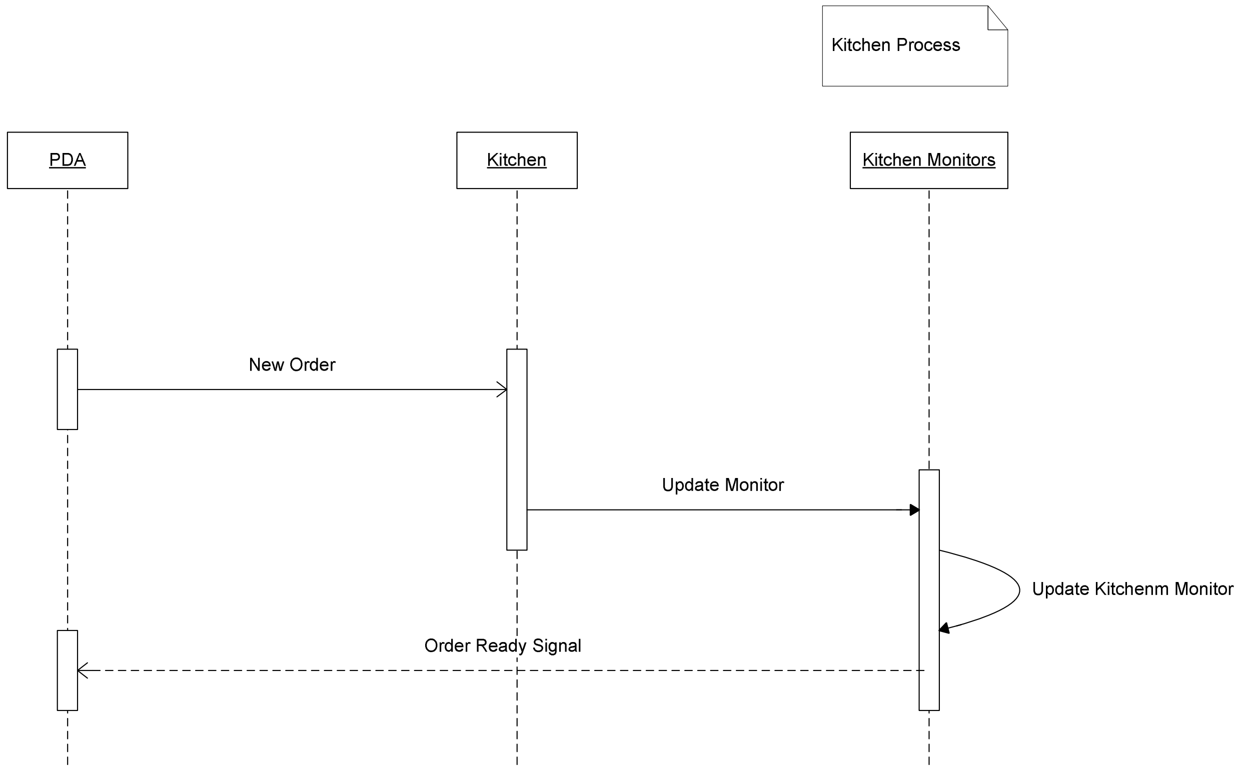

Send Bill
Waiter
PDA User Interface
Securiy Login System
Personal Accounts System
Tables
Cash Desk
ID/Password
Check the User
Check ID/Password
Load Account
Load Personal Account
Open New Table
Open New Table
Check Table Status
Select a Table
Select New Table
Select Table
Close a Table
Close Table
Close Table
Send Bill
Logout Wish
Logout


View Orders
Cashier
Active Order List
Monitor
List Order
Display Order

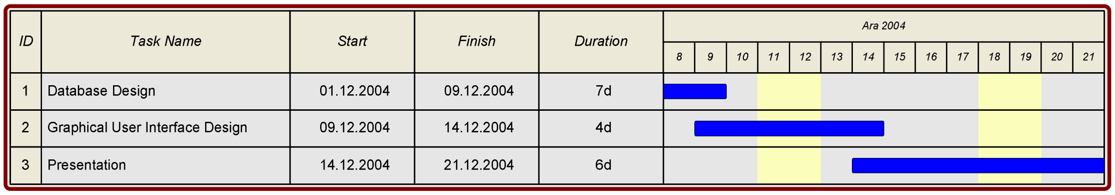

Generate Statistics
Cashier
Statistics
Passive Order List
Monitor
Generate Statistics
Get Orders
Orders
Generate Statistics
Display Graphics


Generate Bill
PDA
User Interface
Bill List
Cashier Desk
Printer
Request Bill
Add to Bill List
Select Bill from queue
Send to Printer
Print


Get EndOfDay
Cashier
User Interface
Passive Order List
Executive System
Printer
Monitor
Get End-Of-Day
Activate List
Send Orders
Execute Orders
Print End-Of-Day
Display End-Of-Day



6. GANTT-CHART DIAGRAM



ID	Task Name	Start	Finish	Duration
1	Database Design	01.12.2004	09.12.2004	7d
2	Graphical User Interface Design	09.12.2004	14.12.2004	4d
3	Presentation	14.12.2004	21.12.2004	6d