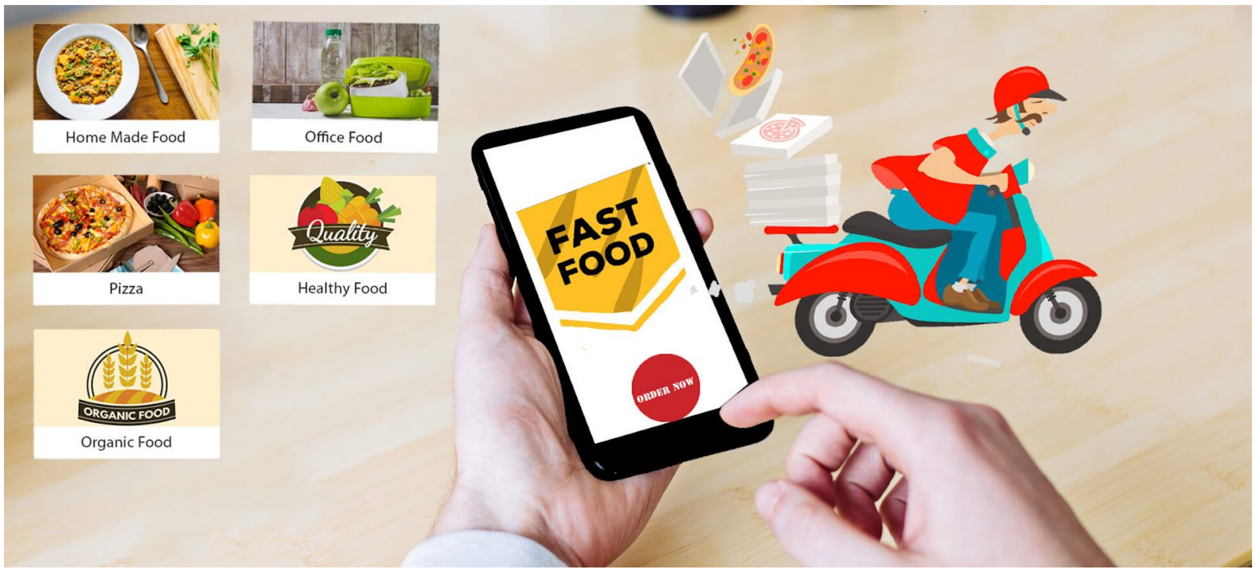


Software Requirements Specification online Restaurant Menu & Ordering System

AMARNAATH 2018506014

AJAI 2018506010

NABEEL 2018506068

CONTENTS :

1 INTRODUCTION

- ❖ *The purpose of this SRS is to outline both the functional and non-functional requirements of the subject RMOS.*
    - ➢ *In addition to said requirements, the document also provides a detailed profile of the external interfaces, performance considerations and design constraints imposed on the subsequent implementation.*
- ❖ *The following section provides an overview of the derived Software Requirements Specification for the subject Restaurant Menu and Ordering System.*
    - ➢ *Subsequently, the scope of the project specified by the document is given with a particular focus on what the resultant software will do and the relevant benefits associated with it.*
- ❖ *To conclude, a complete document overview is provided to facilitate increased reader comprehension and navigation.*
    - ➢ *To this audience group, this SRS should convey and confirm the required functionality and represent a contractual agreement between the involved parties.*

2 Scope

- ➔ *In current formal dining environments, some form of physical static menu is utilised to convey the available food and beverage choices to customers.*
  - ◆ *Said menus are generally paper based and hence impose restrictions on the textual real estate available and the ability a restaurateur has to update them.*
- ➔ *This document specifies the requirements for a restaurant paper menu and ordering replacement strategy to alleviate the problems associated with the current archaic method.*
  - ◆ *Three related concepts are encompassed by the general scope of the Restaurant Menu and Ordering System.*
- ➔ *The first pertains to the replacement of paper-based menus using an electronic format, the second relates to a complementary electronic strategy for the front of house handling of a customer's order and the third surrounds the process of transferring said electronic orders to the kitchen for preparation.*
  - ◆ *It should be noted that while the suggested strategy incorporates the use of various hardware components, the*

primary focus of the presented SRS relates to the constituent software elements.

Overview

- *The Restaurant Menu and Ordering System is a software package to facilitate ordering within a traditional restaurant.*
    - *The customer is able to view the menu, place orders, call the waiter, and organise the final bill through the surface computer interface built into their table.*
        - *Waiters are able to initialise a table for customers, control table functions remotely to assist customers, confirm orders, send orders to food preparation staff and finalise the customer's bill – all through their wireless tablet PC.*
    - *The food staff, with their touch-display interfaces to the system, are able to view orders sent to the kitchen by waiters.*
- *During preparation, they are able to let the waiter know the status of each item, and can send notifications when items are completed, again through the touch- display.*
        - *The system contains full accountability and logging systems, and supports supervisor actions to account for exceptional circumstances, such as a meal being refunded or walked out on.*

BENIFITS

➔ *Greater flexibility in menus, an increase in restaurant productivity and capacity for extensive business auditing are the primary benefits associated with the RMOS.*

- ◆ *Menu updates can be rolled out at any time with no extra labour from printing and distributing new menus, allowing for more dynamic pricing and content changes.*

- ◆ *With the underlying software system taking responsibility for a customer's order throughout its lifecycle, not only is accuracy ensured, but all actions are logged in a database for analysis and accountability of staff.*

- ★ *This allows the company to audit and account staff time, materials and restaurant efficiency, as well as to review exceptional circumstances for future handling.*

OVERALL DESCRIPTION

1) *The following section presents an overall description of the subject RMOS.*
    a) *In particular, the product has been put into perspective through a detailed assessment of the system, user, hardware, software and communication interfaces, memory considerations, operational modes and site adaptation requirements.*
        i) *Further, characteristics of the system's end-users are discussed along with the identified system constraints and assumptions.*
        ii) *To conclude the section, an apportioning of requirements has been outlined.*

Product Perspective

- ❖ *The software described in this SRS is the software for a complete RMOS system.*
    - ➢ *The system merges various hardware and software elements and further interfaces with external systems.*
    - ➢ *Thus, while the software covers the majority of the system's functionality, it relies on a number of external interfaces for*

persistence and unhandled tasks, as well as physically interfacing with humans.

User interfaces

- ➢ *There are three separate user interfaces used by the RMOS software, each related to an interfaced physical hardware device .*
- ➢ *These three user interfaces are the Surface Computer UI, Tablet UI and Display UI.*

Hardware interfaces

- ❖ *There are three external hardware devices used by the RMOS, each related to a user interface .*
  - ➢ *These devices are the surface computers, the wireless tablets and the touch displays.*
  - ➢ *All three devices must be physically robust and immune to liquid damage and stains.*
  - ➢ *The devices must also have good industrial design aesthetics, as they are to be used in place of normal restaurant tables and notepads and will be in direct contact with customers.*
  - ➢ *The devices behave as 'terminals' in the sense that they never have a full system image, do not store data and are not used for the core logic of the system.*
  - ➢ *However, they should be fully capable computers that can use textual data from the server along with local UI/interpretation code to display UI elements and take input. All order and transaction records should be stored on the server, not these computers.*

- *The performance of dumb terminals over an area the size of a restaurant is likely to be unacceptable.*
- *In all three cases, the hardware device takes information from the RMOS and processes the information to display. It also provides user input information to the RMOS.*

Software interfaces

- *The RMOS will interface with a Database Management System (DBMS) that stores the information necessary for the RMOS to operate.*
- *The DBMS must be able to provide, on request and with low latency, data concerning the restaurant's menu, employees (and their passwords) and available dietary requirements.*
- *Additionally, it should take and archive data provided to it by the RMOS.*
- *This data will include records of all orders and transactions (system states and state changes) executed by the RMOS.*
- *The DBMS must store all data such that it can be used for accounting, as well as accountability.*

Assumptions

- *The SRS assumes that none of the constituent system components will be implemented as embedded applications.*
  - *The implication is that the target hardware will provide a capacity for standalone program/application deployment and not require customised embedded firmware to be written.*
  - *It is further assumed that tablet PCs of sufficient processing capability and battery life will be utilised.*
    - *The surface computers employed by the system should facilitate being utilised/left on for extended periods and that they are programmable in the same fashion*

as x86 architecture computers. Finally, it is further assumed that the deployment environment is capable of supporting an wireless network for system communication.

3 REQUIREMENTS

- *The following section presents the complete set of functional and non-functional requirements identified for the subject RMOS.*
- *Functional requirements are listed first, according to their relationship to the overall system, customers, waiters, chefs and supervisors.*
- *The non-functional requirements that pertain to safety, security, the interface, human engineering, qualification, operation, maintenance and performance are subsequently presented.*
- *The functional requirements have been specified using a natural language description and as such, the reader is directed to Section 4 (UML Analysis Models) for further detail.*

3.1 Functional Requirements

This subsection presents the identified functional requirements for the subject RMOS. Initially, general requirements that pertain to the whole system are given. Where possible, subsequent requirements have been demarcated based on their *relevance to the users of the system, that is, customers, waiters, chefs and supervisors.*

3.1.1 General

Requirement Description

1. *G01 A server shall host the RMOS and provide system data processing and storage capability.*
2. *G02 A surface computer shall provide a customer with all customer system functionality.*
3. *G03 A tablet shall provide a waiter/supervisor with all waiter/supervisor system functionality (according to access control).*
4. *G04 A display shall provide a chef with all chef system functionality.*
5. *G05 All system functionality shall be accessible through touch sensitive surface computers, tablets and displays via simple touch gestures.*
6. *G06 A tablet shall be capable of interfacing with a register to facilitate the accurate processing of a payment.*

3.2 Non-Functional Requirements

This subsection presents the identified non-functional requirements for the subject RMOS. The subcategories of non-functional requirements given are safety, security, interfac e, human engineering, qualification, operational and maintenance.

3.2.1 Safety

Table 3.2.1 presents the identified non-functional safety requirements that directly relate to the entire subject RMOS. Requirement Description

1. *F01 The system shall log every state and state change of every surface computer, tablet and display to provision recovery from system failure.*
2. *F02 The system shall be capable of restoring itself to its previous state in the event of failure (e.g. a system crash or power loss).*
3. *F03 The system shall be able to display a menu at all times to facilitate manual order taking should the need arise.*
4. *F04 The system shall utilise periodic 30-second keep-alive messages between tablets and the server to monitor tablet operational status.*
5. *F05 The system shall flag tablets that fail to send timely keep-alive messages as non-operational and disassociate the assigned waiter from the tablet.*

4 UML ANALYSIS MODELS 4.1 Use Cases

This subsection extends upon the functional requirements given in Section 3.1 through the presentation of detailed use cases. To facilitate an unambiguous and clear view of how the end- users interact with the subject RMOS, the actors (end-users) involved in the use cases, a use case diagram and detailed use case descriptions are provided. The use cases that find *representation are Log In, Log Out, Activate Table, Deactivate Table, Accept Order, Deliver Item, Process Bankcard Payment, Process Cash Payment, Abort Meal, Abort Account, Issue Refund, Place Order, Call Waiter, Pay Bill, Accept/Reject Item and Indicate Item Ready.*

4.1.1 Actors

There are four actors in the RMOS, waiter, chef, supervisor and customer. While the waiter, chef and customer actors are base specialisations, the supervisor class inherits from both the waiter and chef actors as a generalisation.

4.1.2 Use case diagram

Figure 4.1.1 presents a use case diagram for the subject RMOS. The key interactions between the end-users of the system and the system itself are depicted.

4.1.3 Use case descriptions

Table 4.1.1 presents the Log In use case description to show the interaction between a waiter and a tablet when logging into the system

Log In Use Case Description

Use Case	*Log In*
Primary Actor	*Waiter*
Goal In Context	*Enable waiter access to the system through a tablet*
Preconditions	*The waiter has a valid username and password and is not already logged in*
Trigger	*The waiter requires access to the system to perform their job*
Scenario	*1) The waiter selects 'Log In' from the tablet menu* *2) The tablet prompts the user for their username and password* *3) The user enters their username and password* *4) The tablet enables access to the system according to access control*

Log Out Use Case Description

Use Case	*Log Out*
Primary Actor	*Waiter*
Goal In Context	*Disable waiter access to the system through a tablet*
Preconditions	*The waiter is already logged in*
Trigger	*he waiter no longer requires access to the system to perform their job*
Scenario	*1) The waiter selects 'Log Out' from the tablet menu 2) The tablet disables access to the system*

Activate Table Use Case Description

Use Case	*Activate Table*
Primary Actor	*Waiter*
Goal In Context	*Activate the surface computer in a table to enable customer functionality*
Preconditions	*The waiter is already logged in*
Trigger	*A new group of customers have been seated at the table*
Scenario	*1) The waiter selects ‘Activate Table’ from the tablet menu* *2) The waiter selects a free deactivated table from those available* *3) The waiter selects the seats to be occupied by customers* *4) The surface computer in the table is activated and an account is created 5) An empty order is automatically created for each customer at the table 6) The waiter is assigned to take care of the table’s account*

Accept Order Use Case Description

Use Case	*Accept Order*
Primary Actor	*Waiter*
Goal In Context	*Accept an order that has been placed by a customer*
Preconditions	*A customer has placed an order*

Trigger	*The waiter chooses to serve the customer*
Scenario	*) The waiter selects 'Take Order' from the tablet menu* *2) The waiter selects a table with pending orders from those assigned to them* *3) The waiter selects a pending order from the selected table* *4) The waiter confirms the order and selects 'Accept' from the tablet menu* *5) The items in the order are sent to the kitchen for preparation* *6) The order is added to the customer's meal (and the table's account)*

Deliver Item Use Case Description

Use Case	*Deliver Item*
Primary Actor	*Waiter*
Goal In Context	*Deliver a ready item to its customer*
Preconditions	*A customer has placed an order*
Trigger	*The waiter chooses to serve the customer*
Scenario	*1) The waiter reads the alert, noting the item and its table/seat number 2) The waiter delivers the item to the customer who ordered it* *3) The waiter marks the item as delivered through the table*

Process Bankcard Payment Use Case Description

Use Case	*Deliver Item*
Primary Actor	*Waiter*
Goal In Context	*Charge a table of customers for their meals taking a collective cash payment*
Preconditions	*Meals have been assigned to the table’s cash payment bill*
Trigger	*The waiter chooses to serve the customer*

Scenario	*1.* *1) The waiter selects 'Bill Table' from the tablet menu* *2. 2) The waiter selects a table with outstanding bills from which the customer's have asked to finalise their account* *3. 3) The waiter selects 'Cash Payment' from the tablet menu* *4. 4) The waiter takes cash from customers and moves to the payment system* *5)The tablet automatically connects with the payment system when in proximity* *6) The waiter processes the payment using the payment system* *7) The meals are marked as paid and disappear from the surface computer 8) The waiter returns to the customers with their change*

Issue Refund Use Case Description

Use Case	*Issue Refund*
Primary Actor	*Supervisor*
Goal In Context	*Refund a customer's payment for a meal*
Preconditions	*A customer has placed an order*
Trigger	*Exceptional circumstances arise and the supervisor must refund the payment*

Scenario	*1) The supervisor confirms that refund should be given* *2) The supervisor selects 'Issue Refund' from the tablet menu* *3) The supervisor selects the meal from a list for the given date* *4) The supervisor selects 'Confirm Refund' from the tablet menu* *5) The details of the meal, the refund and the supervisor are archived 6) The supervisor returns the cost of the meal to the customer*

Accept/Reject Item Use Case Description

Use Case	*Accept/Reject Item*
Primary Actor	*Chef*
Goal In Context	*Notify waiters if an item cannot be prepared*
Preconditions	*The item has not already been accepted or rejected*
Trigger	*Chef proceeds to action the pending items on a kitchen display*

Scenario	*1) The chef checks if the pending item can be fulfilled and rejects or accepts it* *2) An accepted pending item is added to a list of items to prepare* *3) A rejected pending item is removed from the kitchen display and an alert is sent to the waiter assigned to the pending item's accoun*

Class Diagram

Class descriptions

The following subsection presents descriptions for the classes identified for the subject RMOS.

Item

This class represents an item of food or beverage from the restaurant's menu. It exists as a part of either a single order or a single meal, but not both at the same time (an order gets decomposed into the constituent Items and they are added to a meal). An Item that is part of an order may contain a dietary requirement. An Item contains the menu item's name, price, description, status and an image or model to depict the Item on a surface computer or display.

DietaryRequirement

This class represents a customer's dietary requirement (e.g. vegetarian, celiac). It exists as part of either a single Order or a single Item, but not both at the same time. A DietaryRequirement contains both the name and a description of the particular requirement it represents.

Order

This class represents a collection of Items and DietaryRequirements. It is part of a single Meal and maintains information about whether it has been placed or approved. An Order can be in either the placed, cancelled or the approved state. An Order will typically be deleted shortly after it has been approved and its Items added to the customer's Meal, tying the Items in the Order to an Account.

Meal

This class represents a collection of all items ordered by a single customer. It is part of a single Account and maintains the seat number related to the Meal. The total price of a Meal is also maintained.

Account

This class represents the associated Meals and Payments of a Table of customers. It consists of one or more Meals and one or more Payments. An account is related to one Table and is associated with one Tablet.

Table

This class represents a physical table at which a customer may be seated and its integrated surface computer. If the represented table is activated, an Account will be associated with the Table. It contains information about its current mode and its unique identification number. Further, it may be woken (activated), put to sleep (deactivated), placed in ordering mode or placed in billing mode.

Tablet

This class represents a wireless tablet used by staff; it essentially represents a waiter/supervisor logged into the tablet and consequently the system. The class contains information about who is

Alert

This class represents a message sent to a Tablet to alert the waiter/supervisor of an event related to one of the Tables they have been assigned. It maintains a category (e.g. 'item rejected', 'item ready') and a description. An Alert belongs to exactly one Tablet.

Payment

This class represents a payment to be made to the restaurant. It may contain any number of Meals and TipDenominations and is related to exactly one Account. A Payment maintains its total value and the seat number of the paying customer.

BankcardPayment

This class represents an extension of the Payment class for customer payments that are to be paid using a bankcard.

CashPayment

This class represents an extension of the Payment class for customer payments that are to be paid using cash.

TipDenomination

This class represents a tip denomination to be given to a waiter. A TipDenomination belongs to a single Payment and contains information about the denomination value.

USE CASE DIAGRAM :

PURPOSE :

- *Specify the context of a system*
- *Capture the requirements of a system*
- *Validate a systems architecture*
- *Drive implementation and generate test cases*
- *Developed by analysts together with domain experts*

NOTATIONS :

Actor

Use Case

Communication Link

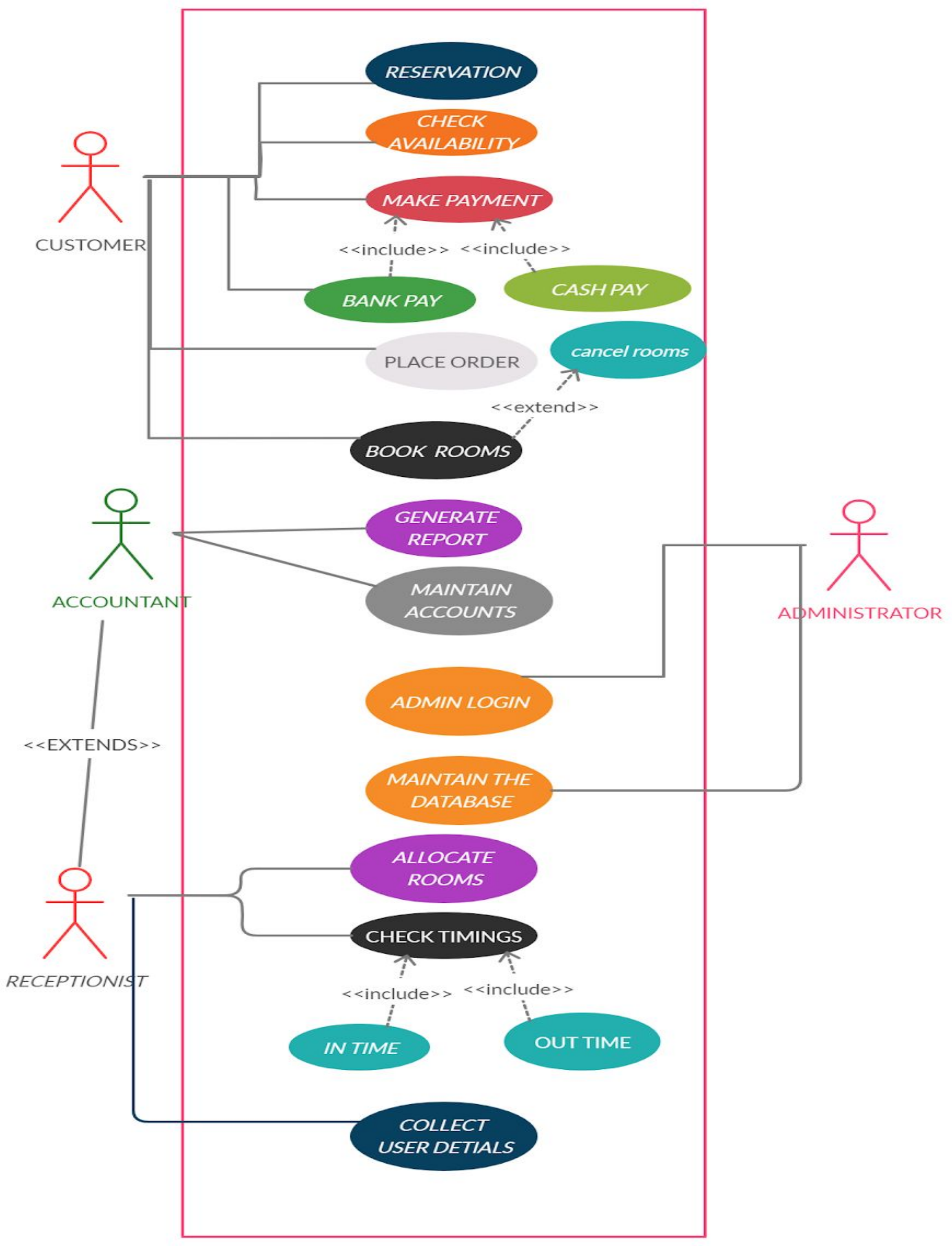
Boundary of system

Use Case Relationship

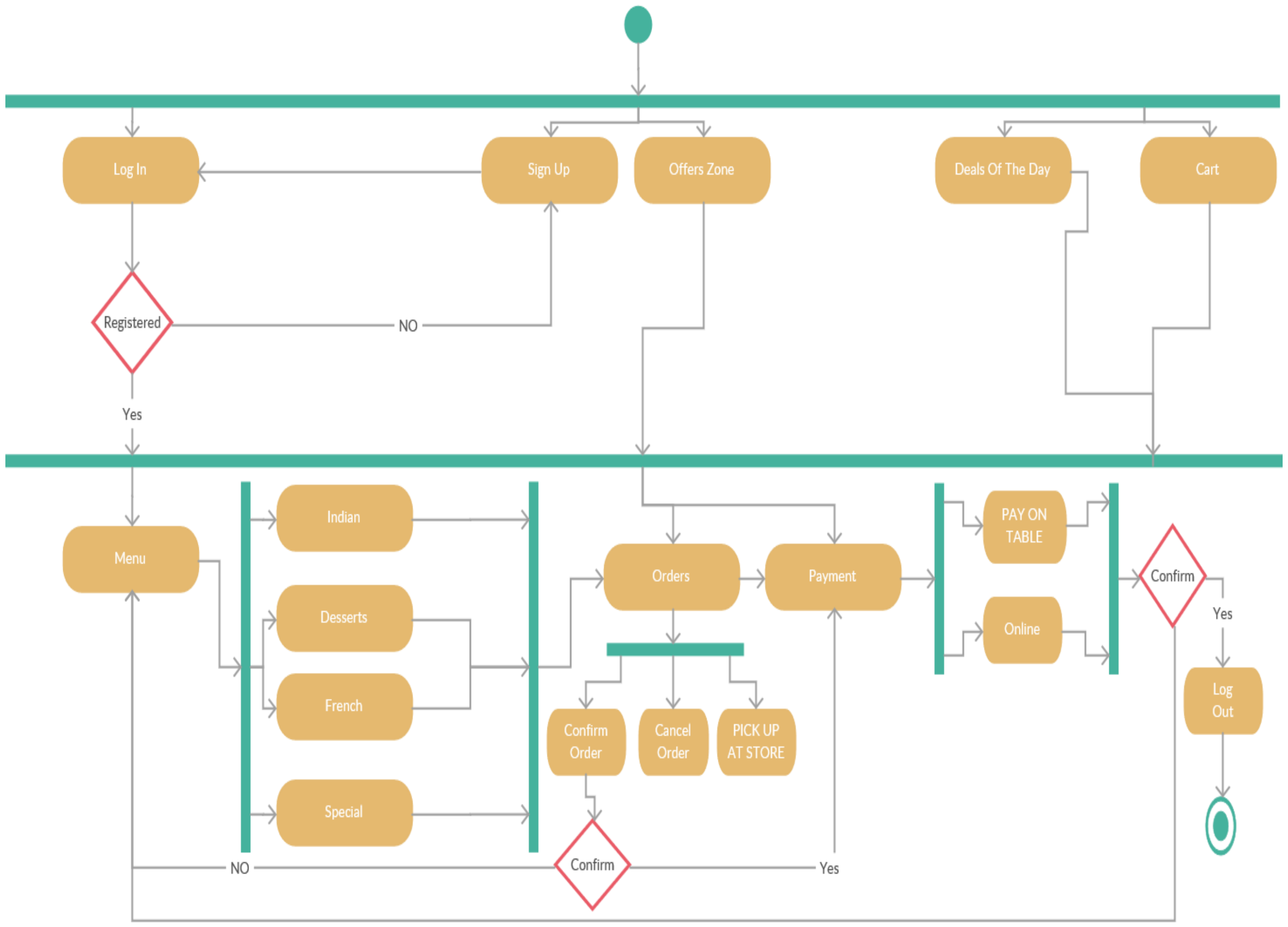
Extends

Include

USE CASE DIAGRAM FOR ONLINE HOTEL BOOKING

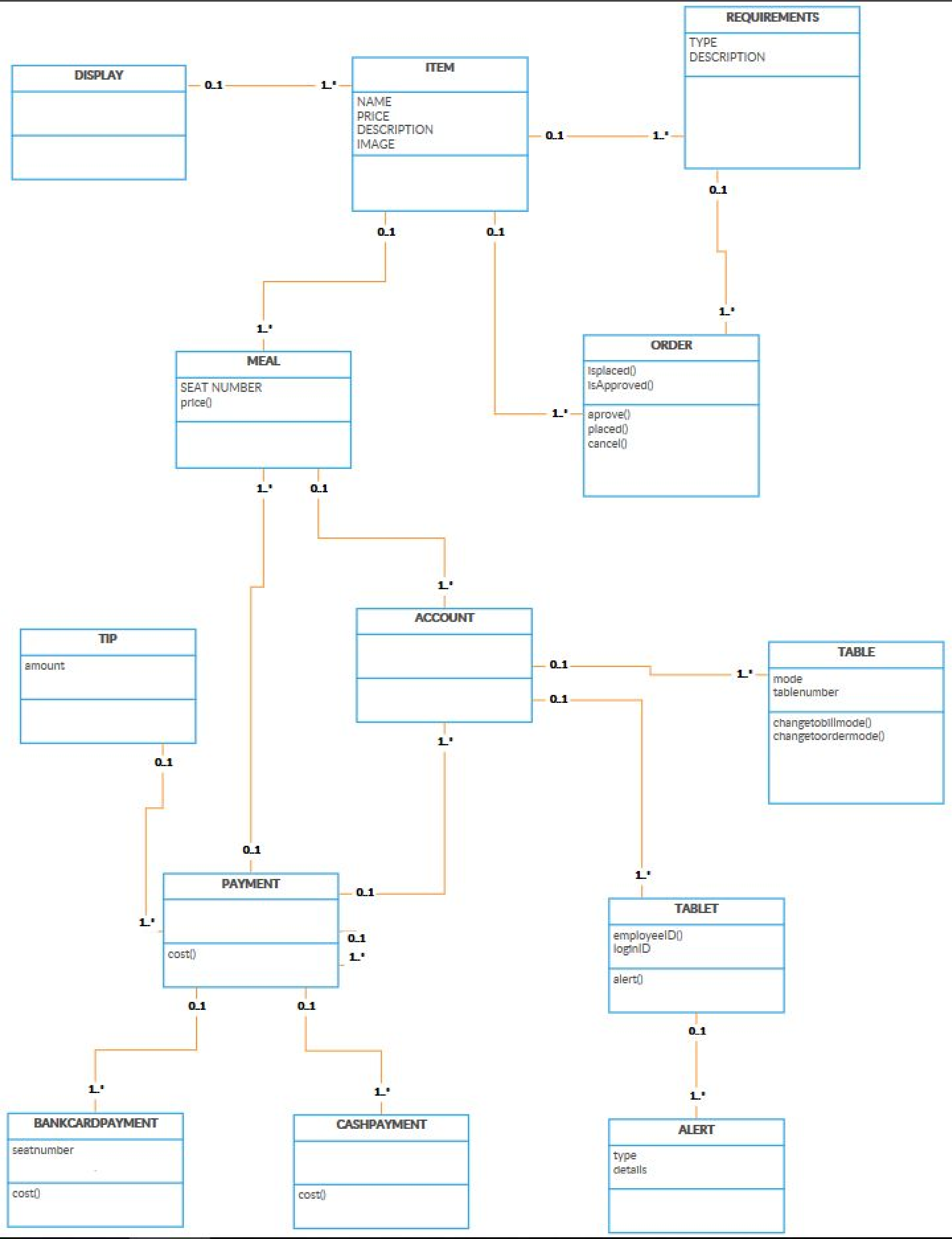

RESERVATION
CHECK AVAILABILITY
CUSTOMER
MAKE PAYMENT
<<include>>
<<include>>
BANK PAY
CASH PAY
PLACE ORDER
cancel rooms
<<extend>>
BOOK ROOMS
ACCOUNTANT
GENERATE REPORT
MAINTAIN ACCOUNTS
ADMINISTRATOR
ADMIN LOGIN
<<EXTENDS>>
MAINTAIN THE DATABASE
ALLOCATE ROOMS
RECEPTIONIST
CHECK TIMINGS
<<include>>
<<include>>
IN TIME
OUT TIME
COLLECT USER DETIALS

Swimlane diagram :

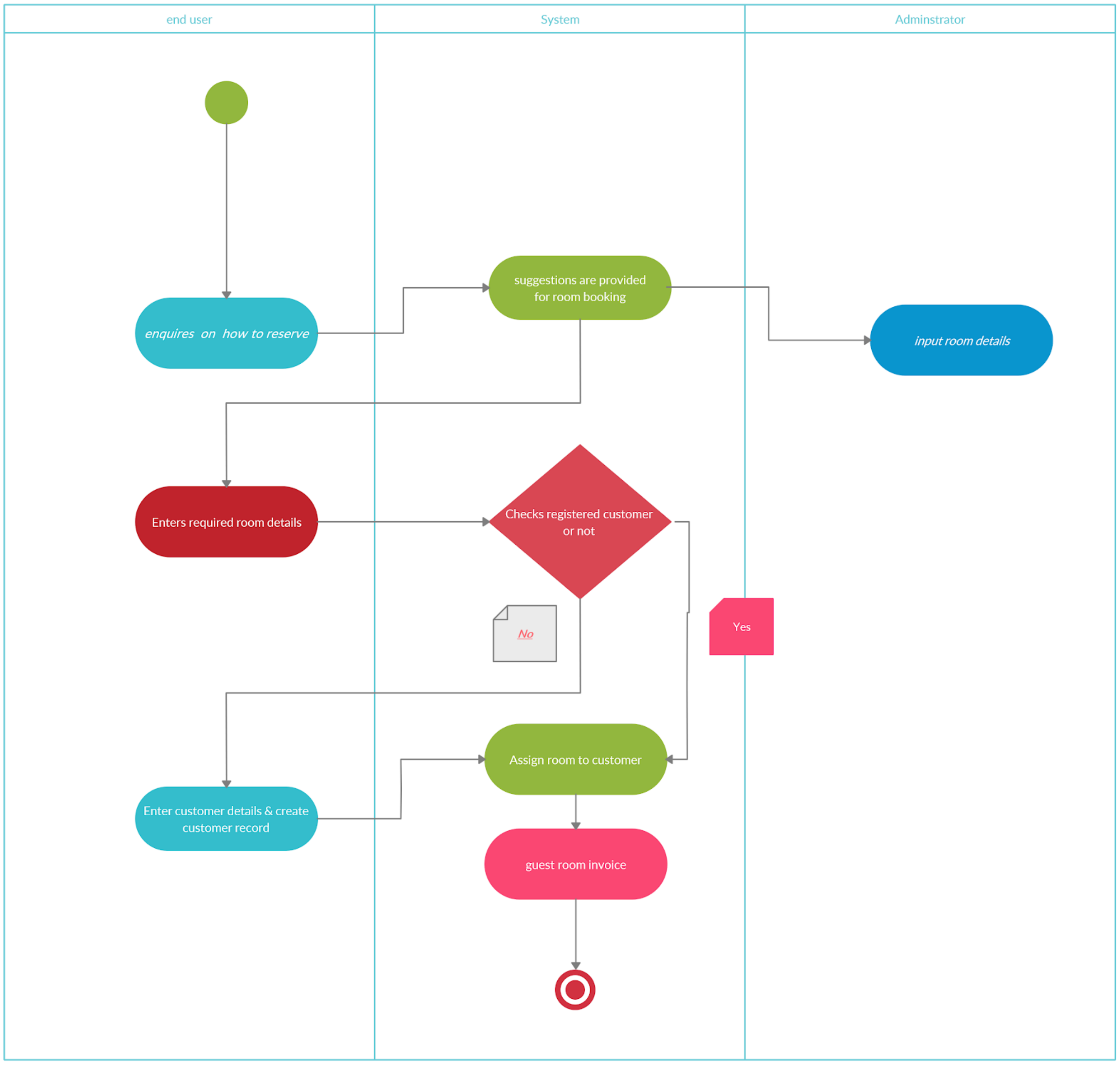


Class diagram :

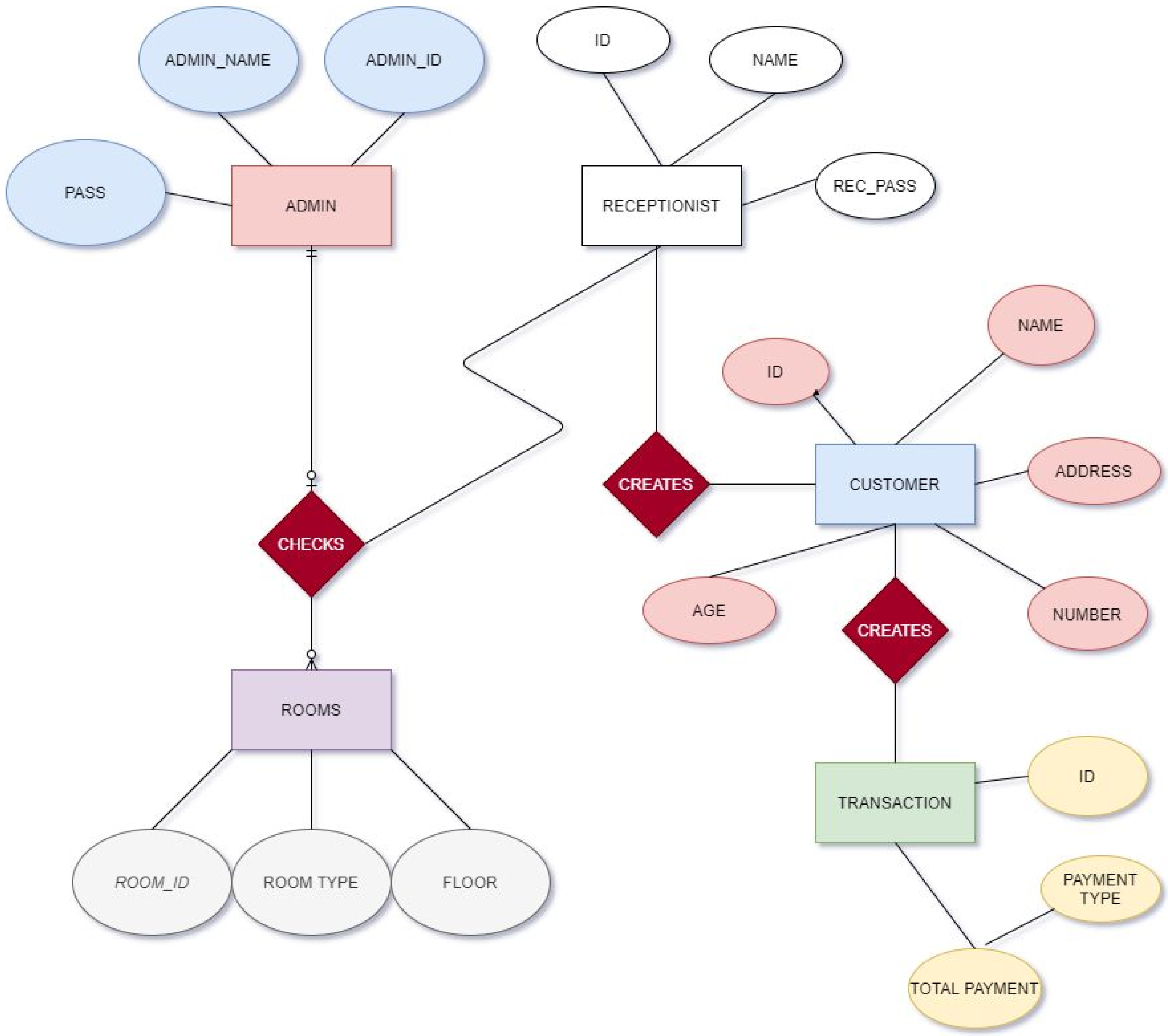
J


DISPLAY
ITEM
NAME
PRICE
DESCRIPTION
IMAGE
REQUIREMENTS
TYPE
DESCRIPTION
MEAL
SEAT NUMBER
price()
ORDER
Isplaced()
IsApproved()
aprove()
placed()
cancel()
ACCOUNT
TIP
amount
TABLE
mode
tablenumber
changetobillmode()
changetoordermode()
PAYMENT
cost()
TABLET
employeeID()
loginID
alert()
BANKCARDPAYMENT
seatnumber
cost()
CASHPAYMENT
cost()
ALERT
type
details
0..1
1..*

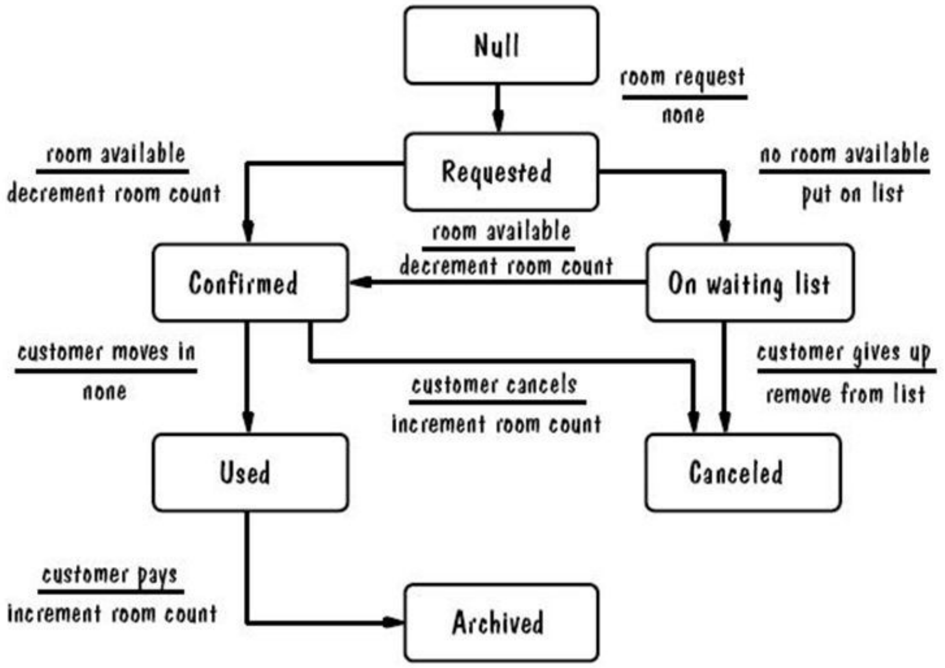
ACTIVITY DIAGRAM :


end user
System
Adminstrator
enquires on how to reserve
suggestions are provided for room booking
input room details
Enters required room details
Checks registered customer or not
No
Yes
Enter customer details & create customer record
Assign room to customer
guest room invoice

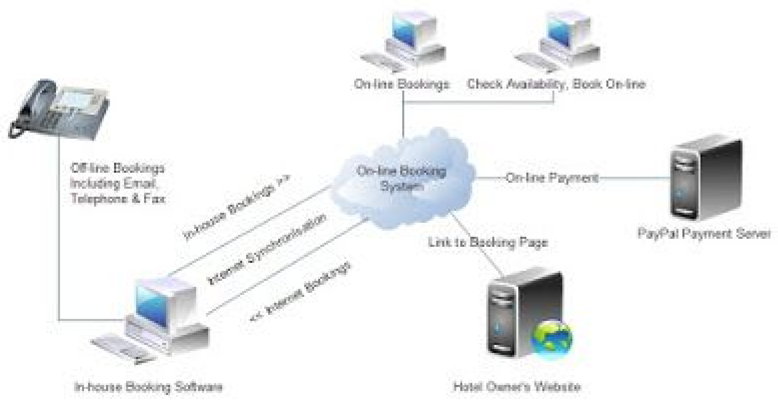
ENTITY RELATIONSHIP DIAGRAM :

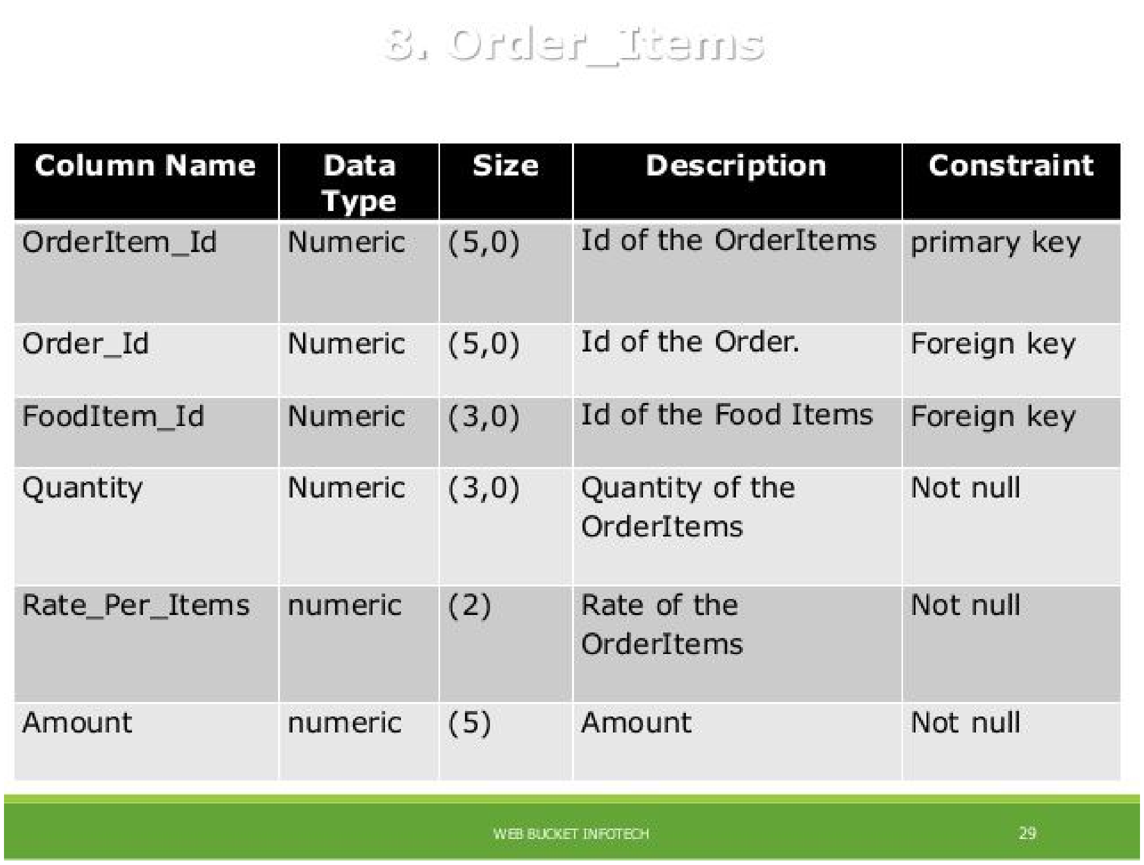


Hotel Reservation State Transition Diagram



Architecture :





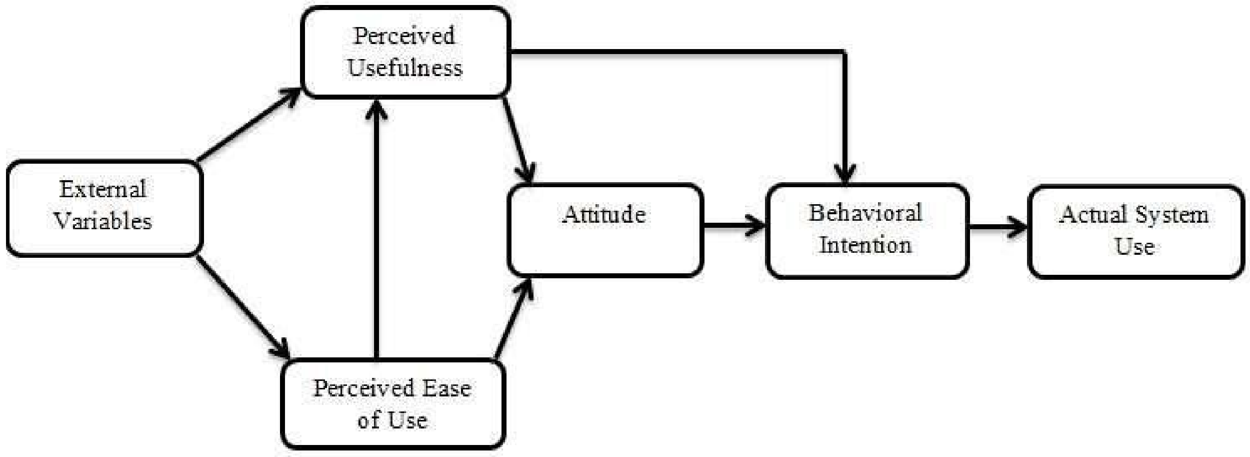
8. Order_Items

Column Name	Data Type	Size	Description	Constraint
OrderItem_Id	Numeric	(5,0)	Id of the OrderItems	primary key
Order_Id	Numeric	(5,0)	Id of the Order.	Foreign key
FoodItem_Id	Numeric	(3,0)	Id of the Food Items	Foreign key
Quantity	Numeric	(3,0)	Quantity of the OrderItems	Not null
Rate_Per_Items	numeric	(2)	Rate of the OrderItems	Not null
Amount	numeric	(5)	Amount	Not null

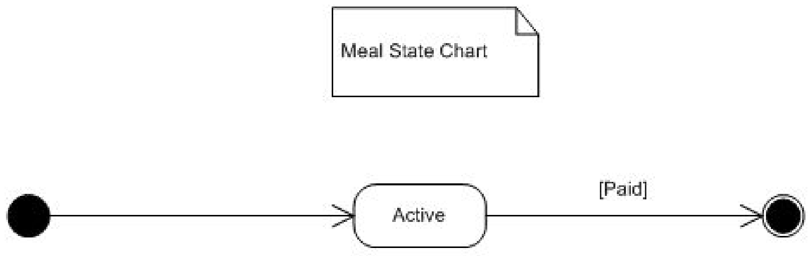
4.BEHAVIOURAL MODELLING :

- *A BEHAVIOURAL MODEL SHOWS THE INTERACTIONS BETWEEN OBJECTS TO PRODUCE SOME PARTICULAR SYSTEM BEHAVIOUR THAT IS SPECIFIED AS A USE-CASE.*

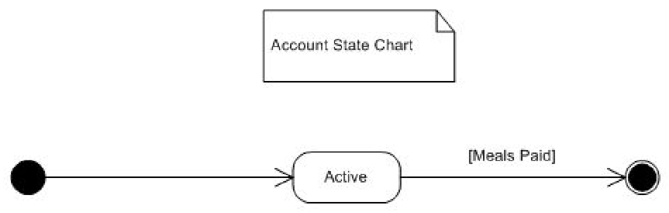
★ CREATING A BEHAVIOURAL MODEL



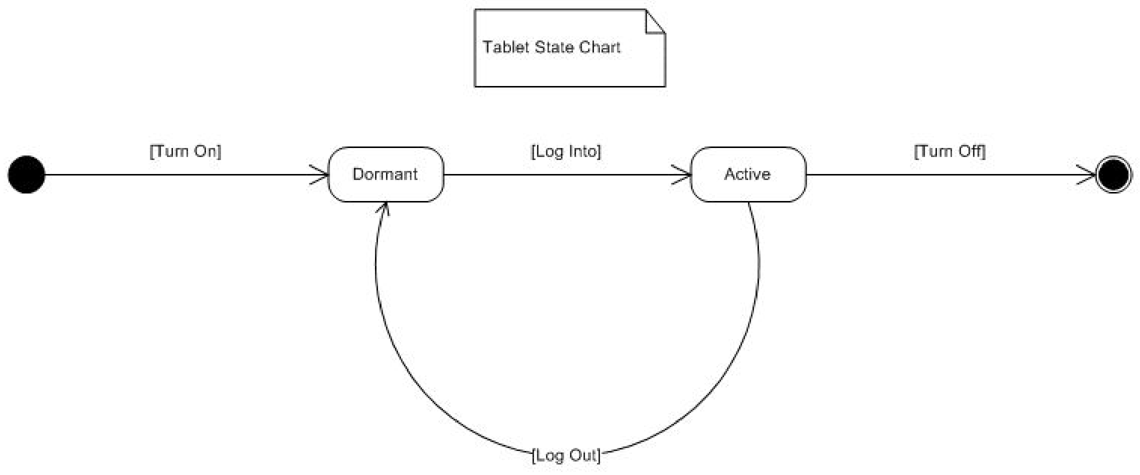
★ STATE REPRESENTATION AND DIAGRAMS



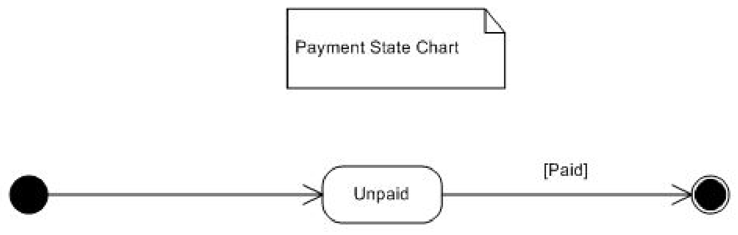
Meal Class Statechart Diagram



Account Class Statechart Diagram



Tablet Class Statechart Diagram

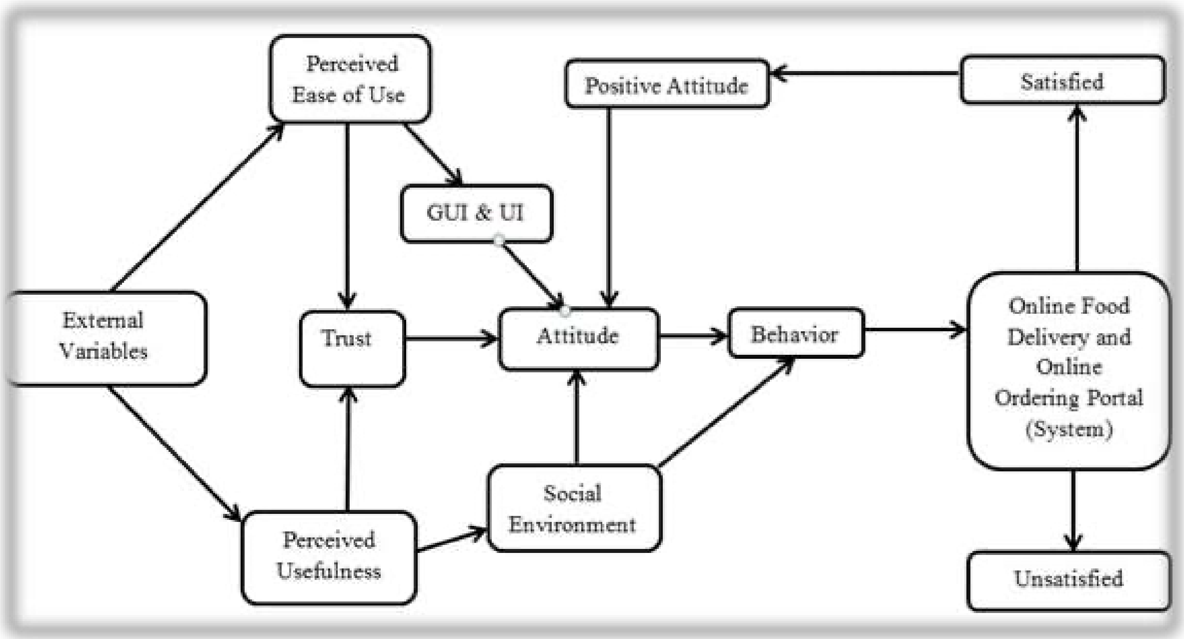


★ ANALYSIS PATTERN

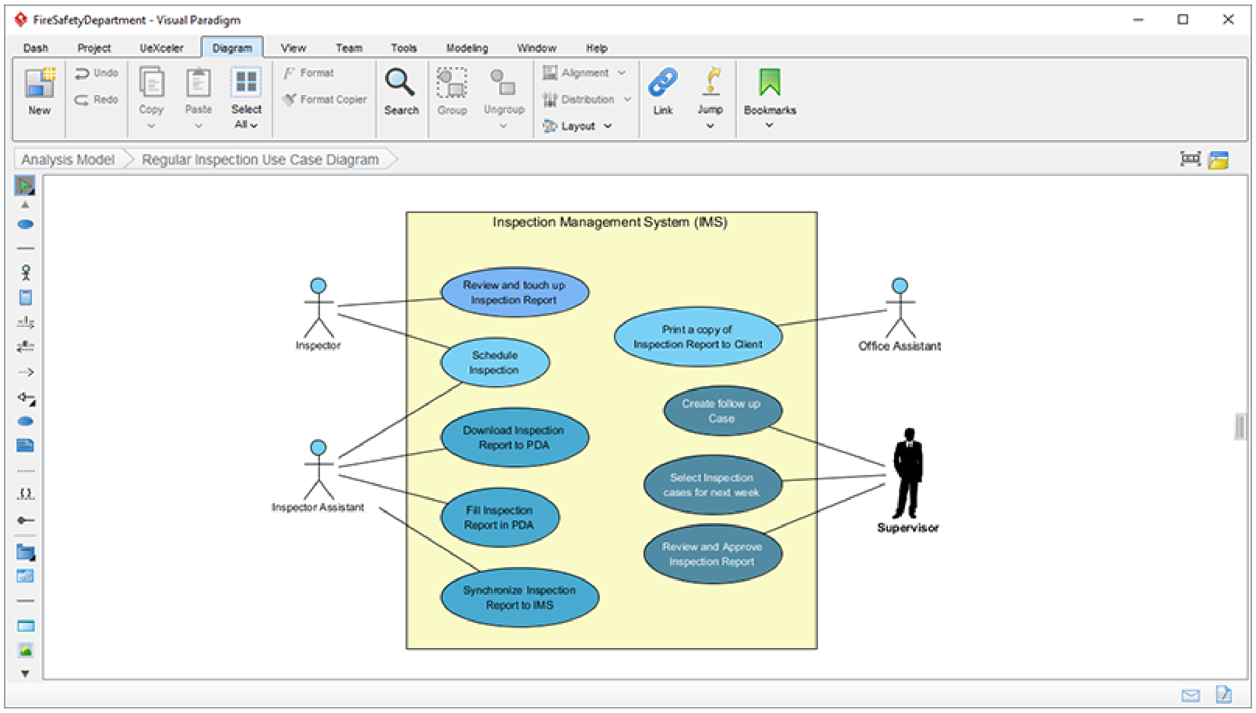
Objectives :

1. ***To quicken the requirements analysis phase by providing reusable analysis models with the description of both advantages and limitations.***

2. ***To suggest several design patterns and feasible solutions to common problems in order to help the software designer translate an analysis model into a design model.***



Functional requirement model


FireSafetyDepartment - Visual Paradigm
Dash
Project
UeXceler
Diagram
View
Team
Tools
Modeling
Window
Help
New
Undo
Redo
Copy
Paste
Select All
Format
Format Copier
Search
Group
Ungroup
Alignment
Distribution
Layout
Link
Jump
Bookmarks
Analysis Model
Regular Inspection Use Case Diagram
Inspection Management System (IMS)
Review and touch up Inspection Report
Schedule Inspection
Download Inspection Report to PDA
Fill Inspection Report in PDA
Synchronize Inspection Report to IMS
Print a copy of Inspection Report to Client
Create follow up Case
Select Inspection cases for next week
Review and Approve Inspection Report
Inspector
Inspector Assistant
Office Assistant
Supervisor

5.FLOW BASED MODELLING :

- ❖ *Flow based modelling describes or provides an abstract of how the data objects will be when they are processed at different stages*
- ❖ *It is used to represent the flow of the system*

★ IDENTIFYING DATA ELEMENTS

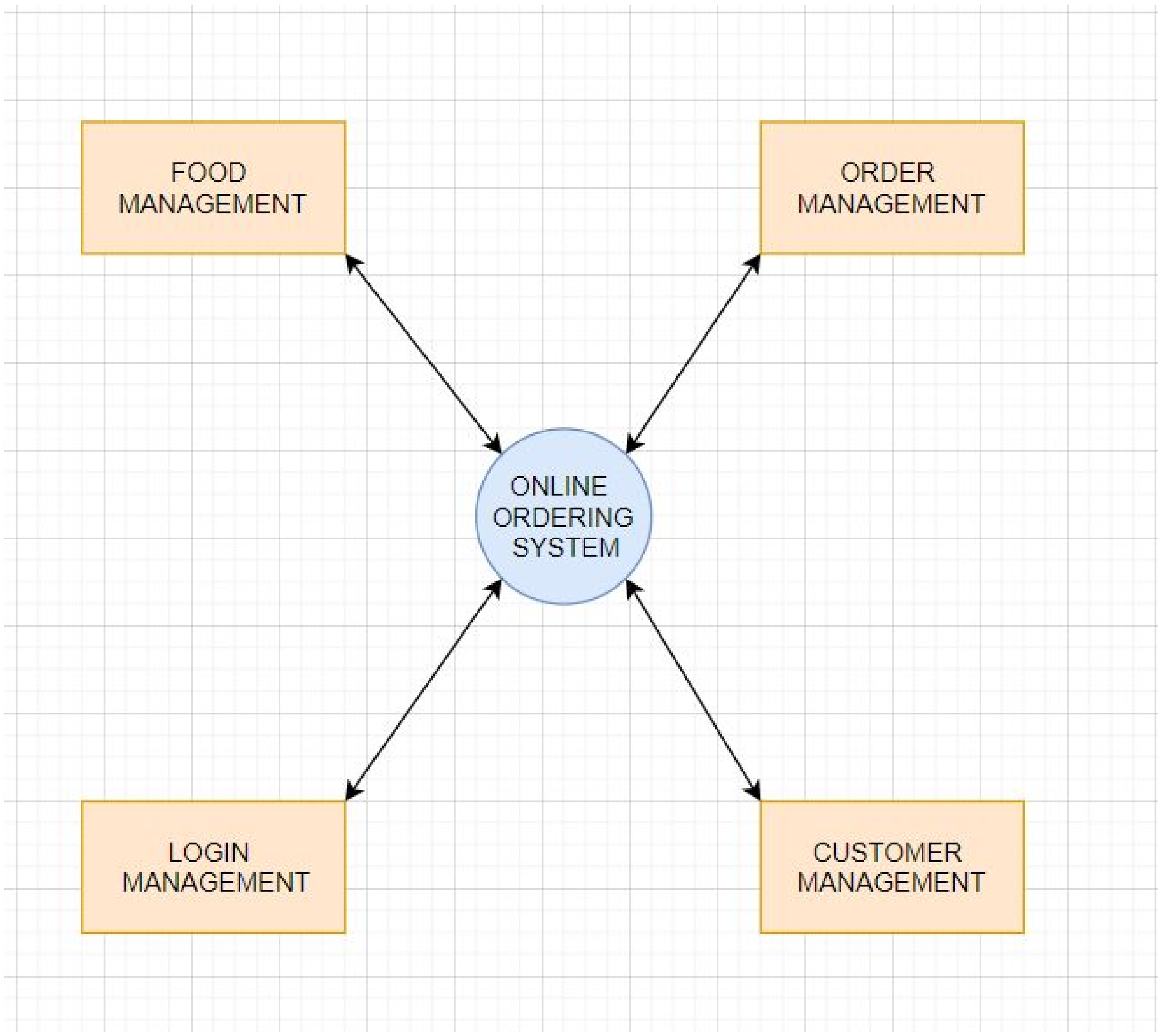
1. *Entity*
2. *Process*
3. *Data center*
4. *Data flow*

❖*Entity = USER , ADMIN*

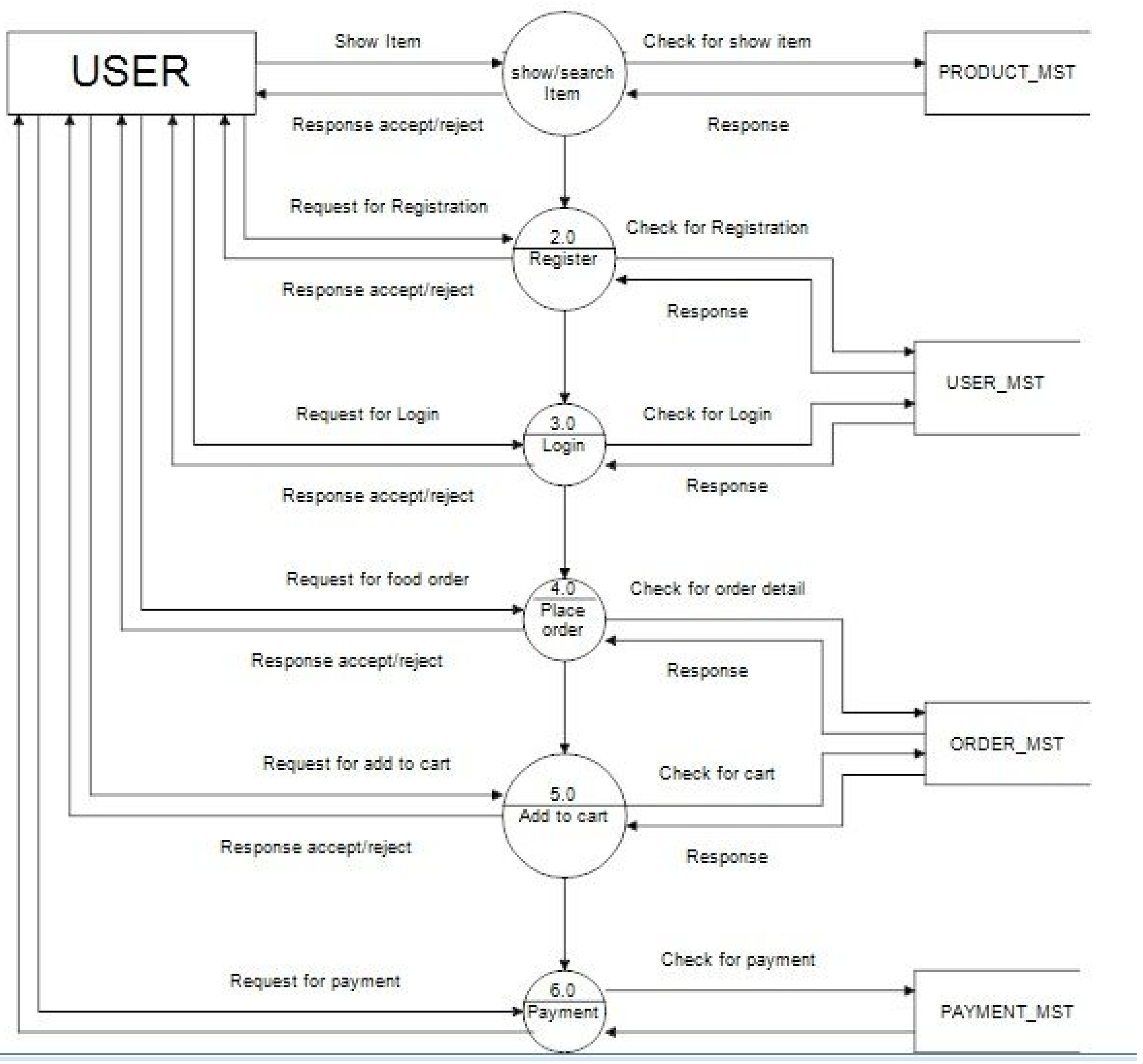
❖*Process = software system*

★ DATA FLOW DIAGRAM :

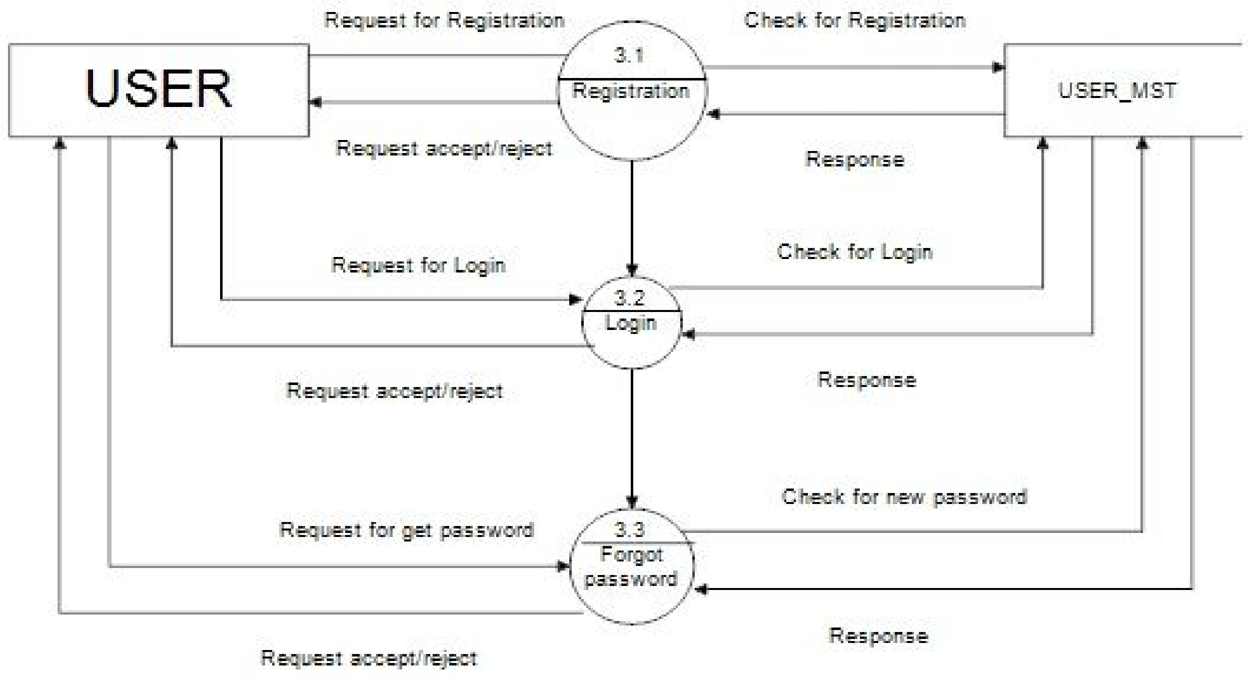
DFD 0 :



DFD 1 :


USER
Show Item
Response accept/reject
show/search Item
Check for show item
Response
PRODUCT_MST
Request for Registration
Response accept/reject
2.0 Register
Check for Registration
Response
USER_MST
Request for Login
Response accept/reject
3.0 Login
Check for Login
Response
Request for food order
Response accept/reject
4.0 Place order
Check for order detail
Response
ORDER_MST
Request for add to cart
Response accept/reject
5.0 Add to cart
Check for cart
Response
Request for payment
6.0 Payment
Check for payment
PAYMENT_MST

DFD 2 :



6.DESGIN MODELLING :

DESIGN CONCEPTS

- ➢ 1. **Abstraction**
- ➢ 2.**Information hiding**
- ➢ 3. **Functional independence**
- ➢ 4.**Refactoring**
- ➢ 5.**Modularity**

1.ABSTRACTION :

- ★ ***A solution is stated in large terms using the language of the problem environment at the highest level abstraction.***
- ★ ***The lower level of abstraction provides a more detail description of the solution.***
- ★ ***A sequence of instruction that contain a specific and limited function refers in a procedural abstraction.***
- ★ ***A collection of data that describes a data object is a data abstraction.***

2.INFORMATION HIDING:

Modules must be specified and designed so that the information like algorithm and data presented in a module is not accessible for other modules not requiring that information.

3.FUNCTIONAL INDEPENDENCE :

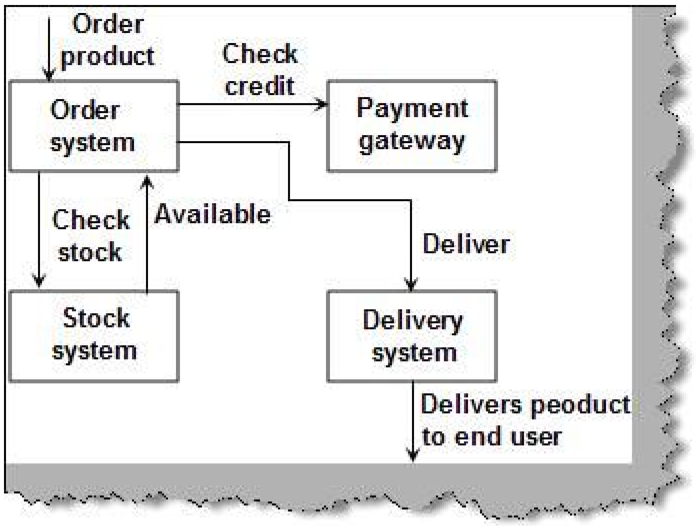
- ★ ***The functional independence is the concept of separation and related to the concept of modularity, abstraction and information hiding.***
- ★ ***The functional independence is accessed using two criteria i.e Cohesion and coupling.***

4.REFACTORING :

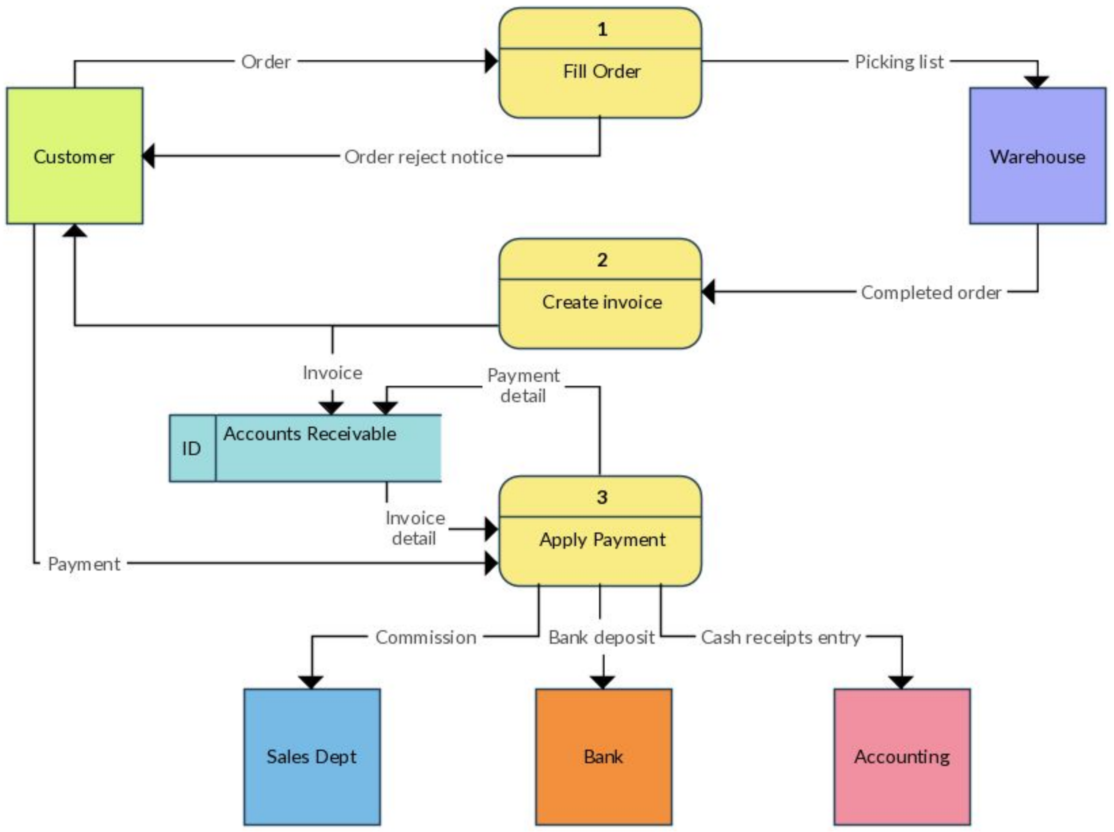
- ***It is a reorganization technique which simplifies the design of components without changing its function behaviour.***
- ***Refactoring is the process of changing the software system in a way that it does not change the external behaviour of the code still improves its internal structure***

5.MODULARITY :

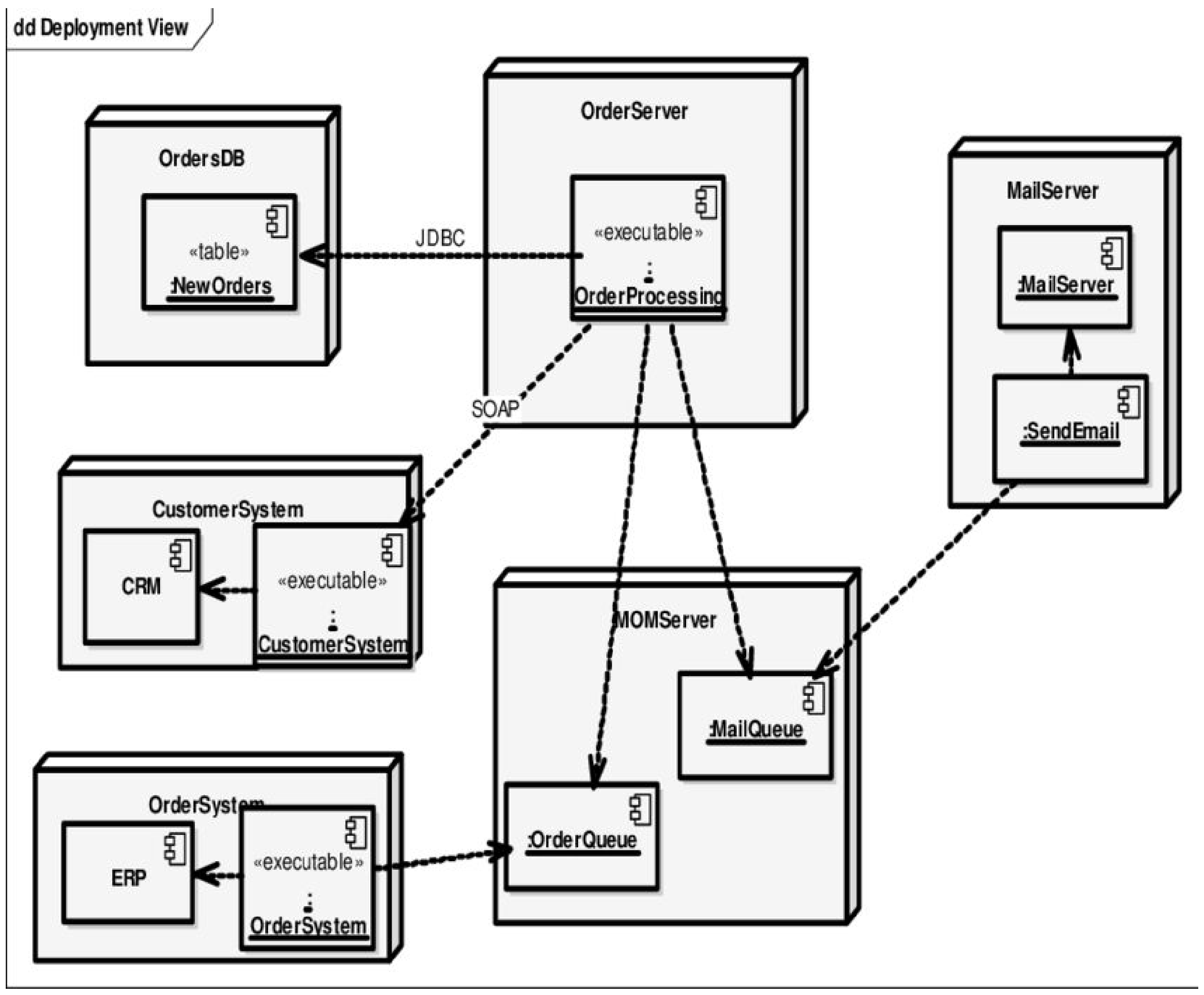
- ***A software is separately divided into name and addressable components. Sometime they are called as modules which integrate to satisfy the problem requirements.***
- ***Modularity is the single attribute of a software that permits a program to be managed easily.***



★ COMPONENT LEVEL DESIGN ELEMENTS



★ DEVELOPMENT DESIGN ELEMENTS



7.CONSTRUCTIVE COST MODEL (COCOMO MODEL)

ONCE A PROCESS MODEL IS FINALEZED FOR A SOFTWARE DEVELOPMENT, SOFTWARE PROJECT MANAGEMENT BEGINS WITH A SET OF ACTIVITIES THAT ARE COLLECTIVELY CALLED PROJECT PLANNING.

PROJECT PLANNING ENCOMPASSESS FIVE MAJOR ACTIVITIES:

- ***ESTIMATION***
- ***SCHEDULING***
- ***RISK ANALYSIS***
- ***QUALITY MANAGEMENT PLANNING***
- ***CHANGE MANAGEMENT PLANNING***

Cocomo (Constructive Cost Model) is a regression model based on LOC, i.e number of Lines of Code.

The key parameters which define the quality of any software products, which are also an outcome of the Cocomo are primarily Effort & Schedule:

- ***Effort: Amount of labor that will be required to complete a task. It is measured in person-months units.***
- ***Schedule: Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put. It is measured in the units of time such as weeks, months.***

According to boehms' definition , this project "online food delivery " comes Under semidetached category !

semi-detached ***– A software project is said to be a Semi-detached type if the vital characteristics such as team-size, experience, knowledge of the various programming environment lie in between that of organic and Embedded. The projects classified as Semi-Detached are comparatively less familiar and difficult to develop compared to the organic ones and require more experience and better guidance and creativity. Eg: Compilers or different Embedded Systems can be considered of Semi-Detached type.***

The Intermediate COCOMO formula now takes the form:

$$E = (a(KLOC)^b) * EAF$$

where, E represents the estimation

a and b are constants ,

EAF is effort adjustment factor

A is taken to be 3.0 and b is taken to be 1.12

Classification of Cost Drivers and their attributes:

- ***Product***
- ***Hardware***
- ***Personal***
- ***Project***

Hence EAF IS CALCULATED BY MULTIPLYING THE COST DRIVER ATTRIBUTES AND THEN USED FOR ESTIMATION

EAF = required software reliabaility * size of application database * complexity of the product*runtime performance constraints*memory constraints * volatility of the virtual machine * environemnt * required turnaruound time * analyst capability * applictations experience * software engineer capability * virtual machine experience * programming language experience * application of software engineering * methods * use of software tools * required development schedule .

EAF = 1 * 0.75 * 1 * 0.75 *1 * 1* 1 * 0.75 * 1.25 * 1 * 1 * 1 * 1.12 * 1

=***0.590625***

HENCE E = 3 * (7)^1.12 * 0.590625

THEREFORE , ESTIMATED VALUE = 15.6654503826

CONCLUSION :

Therefore, conclusion of the proposed system is based on user's need and is user centered. The system is developed in considering all issues related to all user which are included in this system. Wide range of people can use this if they know how to operate android smart phone. Various issues related to Mess/Tiffin Service will be solved by providing them a full fledged system. Thus, implementation of Food Ordering system is done to help and solve one of the important problems of people.

Based on the result of this research, it can be concluded: It helps customer in making order easily; It gives information needed in making order to customer. The Food order application made for restaurant and mess can help restaurant and mess in receiving orders and modifying its data and it is also made for admin so that it helps admin in controlling all the Food system. With food ordering system, a restaurant and mess menu online can be set up and the customers can easily place order. Also with a food menu, tracking the orders is done easily, it maintain customer's database and improve the food delivery service. The restaurants and mess can even customize on restaurant menu and upload images easily, potential customers can easily access it and place order at their convenience. Thus, an automated food ordering system is presented with features of feedback and wireless communication. The proposed system would attract customers and adds to the efficiency of maintaining the restaurant and mess ordering and billing sections.

Scope of the proposed system is justifiable because in large amount peoples are shifting to different cities so wide range of people can make a use of proposed system.