Food Order App

Rivier University

CS552BH1: Object-Oriented System Analysis & Design

Summer 2016, Term-1

Submitted to: Brian Lopez

Submitted by:

Chaitrali Jayantilal Doshi

Mili Bhargava

Vineel Sai Kumar Rampally

Pavan Koppula

1 Executive Summary

1.1 Introduction

Food Order App is a mobile application for ordering food online from various restaurants. This platform will provide a convenient way to browse through the different restaurant menus and order without going to their individual sites.

1.2 Goal

The goal of this application is to avoid user's efforts by providing single platform to browse and order from nearby restaurants.

1.3 Overview

- A user will have an account for the mobile app service.
- It will have menus from restaurants that have subscribed to this service.
- A user can browse through different menus and place an order.
- The customer has an option to pay the bill using the app itself or pay on delivery.
- The order will be placed at a restaurant nearest to the customer and the subscribed delivery service will be notified. For delivering food charges will be applied. Also User has option for Order to pick up or order to Delivery.
- An agent from the delivery service will pick up the order and deliver it to the customer.

Table of Contents

Food Order App........1
1 Executive Summary........2
1.1 Introduction........2
1.2 Goal........2
1.3 Overview........2
2 Project Plan........5
2.1 Implementation plan........5
2.2 Testing Plan........6
3 Functional Specification........7
3.1 Requirements........7
3.2 Assumption and constraints........7
3.3 Use Case Diagram........8
3.4 Activity Diagram........14
3.5 Sequence Diagram........16
3.6 Class Diagram........17
3.7 Object Diagram........18
3.8 State Machine Diagram........19
4 Functional Test Plan........20
4.1 Login test case for both User and Restaurant Panel........20
4.2 Test Case for Admin Panel........20
4.3 Test Case for User Panel........21
4.4 Test Case for Restaurant Panel........22
5 System Design Specifications........23
5.1 System Physical Architectural........23
5.2 Package Diagram........24

5.3 UI Design 25

5.4 Window Navigation Diagram 27

6 Database Structure 28

6.1 Table Structure 28

6.2 ER Diagram 29

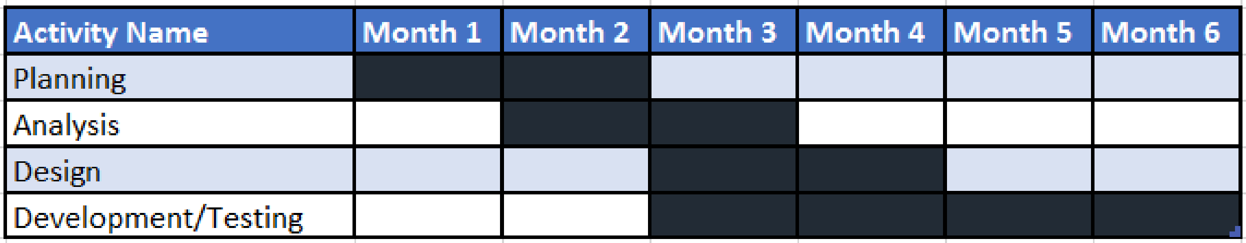
7 System Integration Test Plan 30

8 Issues for future studies 31

9 Conclusion 31

10 References 31

2 Project Plan



Activity Name	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6
Planning	■	■				
Analysis		■	■			
Design			■	■		
Development/Testing			■	■	■	■

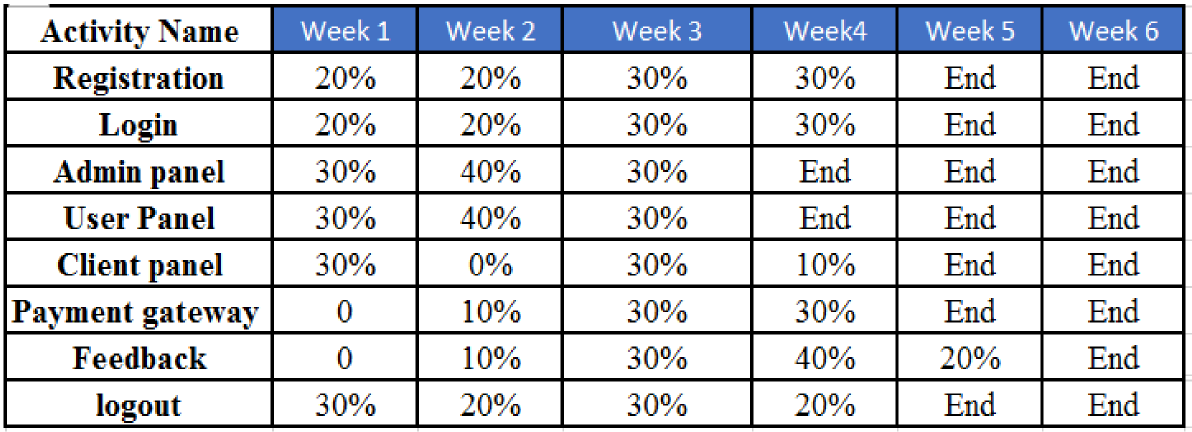
Above diagram shows estimated project plan.

2.1 Implementation plan

Food Order app will have the majorly three modules

i. Admin (System Admin)
ii. User (Customer)
iii.Client (Restaurant Owner)

These three modules have different types of the specification and requirements. Clients and User have features common as the registration and login and forget password other requirements are changed based on the type of login. If the login person is Admin, then the person has the features like approval orders. If the login person is User then he has the features like search store, create order, bill for order, get delivered give feedback of the Restaurants. The following table illustrates that how much time it will take to the complete the each and every requirement. This task estimation is only to implement the requirements not for quality analysis. Here I will describe if three people are working parallel.

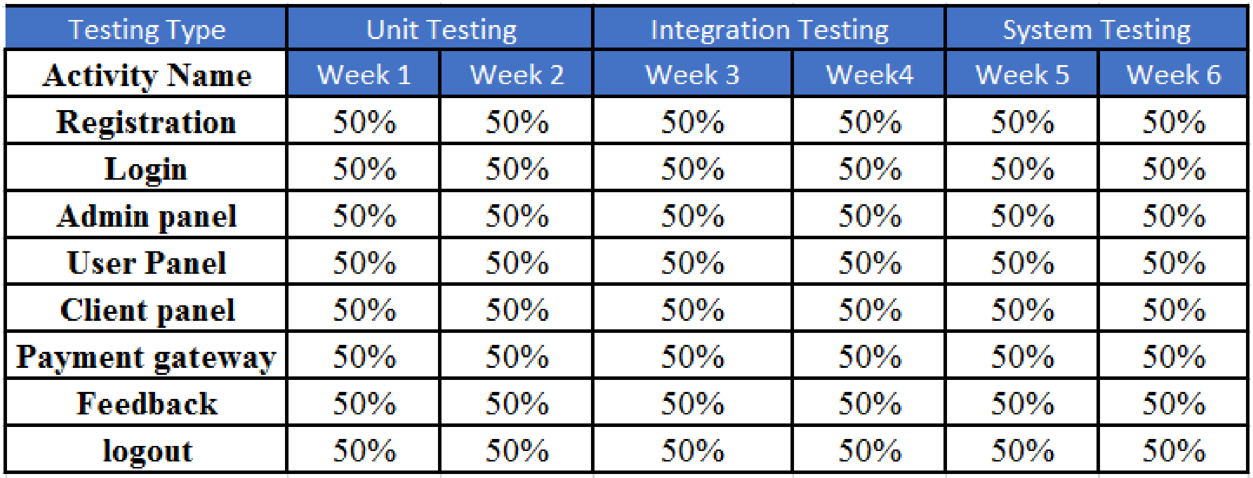


Activity Name	Week 1	Week 2	Week 3	Week4	Week 5	Week 6
Registration	20%	20%	30%	30%	End	End
Login	20%	20%	30%	30%	End	End
Admin panel	30%	40%	30%	End	End	End
User Panel	30%	40%	30%	End	End	End
Client panel	30%	0%	30%	10%	End	End
Payment gateway	0	10%	30%	30%	End	End
Feedback	0	10%	30%	40%	20%	End
logout	30%	20%	30%	20%	End	End

The above table gives a clear idea about our starting and end of the each and every requirement.

2.2 Testing Plan

Testing plan will include various phases as below,



Testing Type	Unit Testing		Integration Testing		System Testing	
Activity Name	Week 1	Week 2	Week 3	Week4	Week 5	Week 6
Registration	50%	50%	50%	50%	50%	50%
Login	50%	50%	50%	50%	50%	50%
Admin panel	50%	50%	50%	50%	50%	50%
User Panel	50%	50%	50%	50%	50%	50%
Client panel	50%	50%	50%	50%	50%	50%
Payment gateway	50%	50%	50%	50%	50%	50%
Feedback	50%	50%	50%	50%	50%	50%
logout	50%	50%	50%	50%	50%	50%

3 Functional Specification

3.1 Requirements

i. **User Registration:** The user will have to create an account for using the app. For creating an account user will need to create User-Id and Password. Once the user has created an account, then he/she can order food from available restaurants menu's.

ii. **Payment Method:** The user has multiple payment option like Pay online and Cash on Delivery. For paying online, you can either pay with a Credit card or Debit Card. Also card information can be stored for future use on request.

iii. **Restaurants Registration:** Restaurants will have to fill out registration form, which will ask for Name of restaurant, Its menu and other information for ordering food from that restaurant.

iv. **Delivery System:** Once the order has been placed, the notification will go to the restaurant as well as Delivery Services. Delivery system will notify their nearest driver to pick food from the restaurant and deliver it to the customer. If payment method is Cash on Delivery, then Driver will collect money as per receipt.

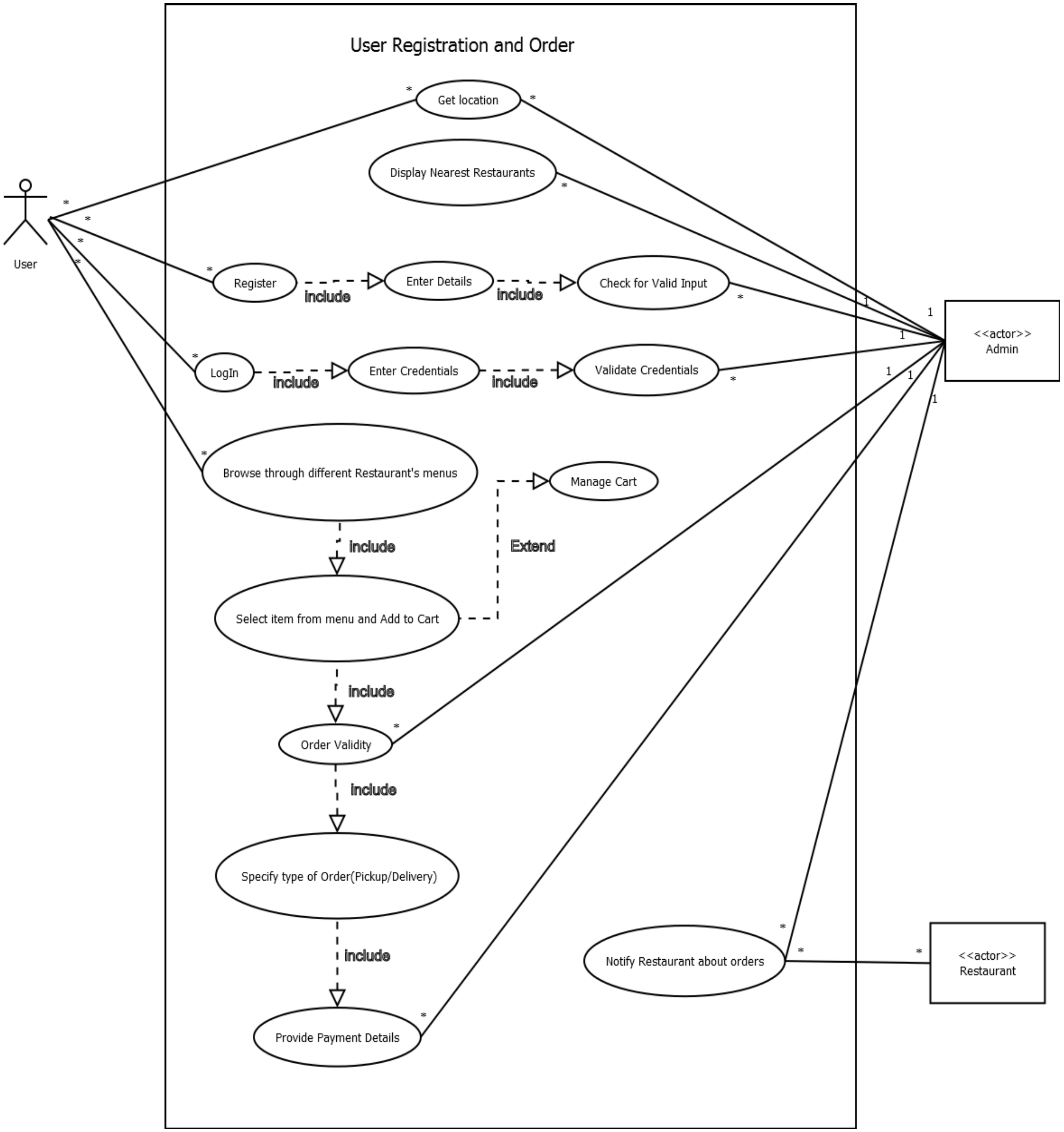
3.2 Assumption and constraints

- The app should be available on the phone.
- Restaurant will be registered for the app service.
- Delivery service has a contract with the App service and restaurants.
- The selected restaurants should be within the range of delivery service.
- The order cancelation is solely at the discretion of the restaurant.
- The order should be placed at least 30 minutes before restaurant closing time.

3.3 Use Case Diagram

This will include 3 use case diagrams for the system.

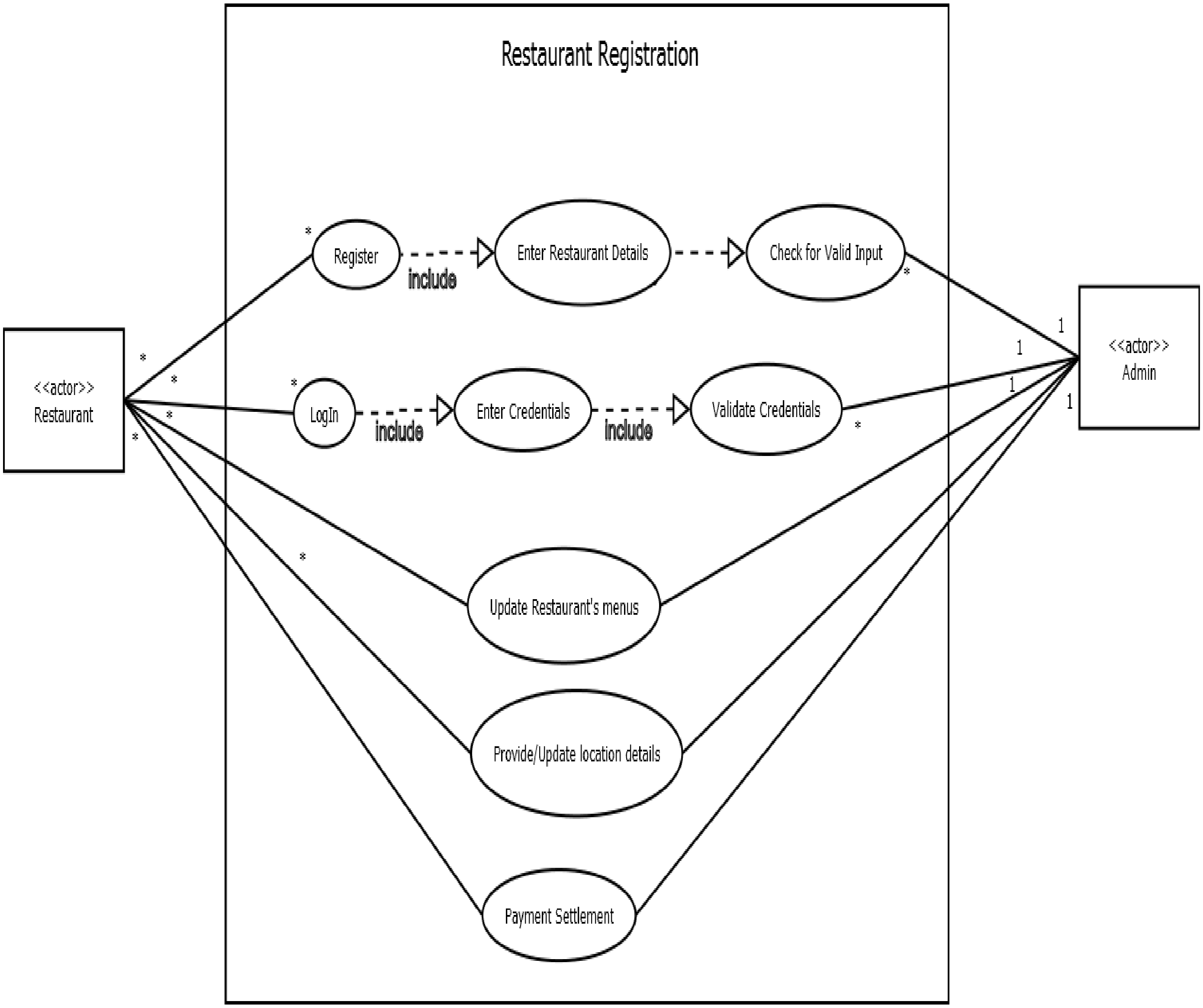
1. User Registration and Order



Use Case Description for User Registration & Order:

Use Case Name:	User Registration & Order
ID:	UC_1
Actors:	User, Admin, Restaurant
Description:	The user logs in using this app and orders food from the nearest restaurant.
Trigger:	The user must be an existing one or properly registered in the case of new user to place the order.
Relationships:	Association: User, System, App
Normal Flow of Events:	1. The user will be displayed the nearest restaurants based on his/her location. 2. A new user will register on the app or an existing user will log in to order food. 3. The credentials will be validated by the system. 4. User will select an item from menu. 5. Items will be added or deleted from the cart. 6. User will provide payment details for the transaction. 7. The order will be validated by the system; *first*, whether the order is placed at least 30 minutes before restaurant closing time; *second*, the type of order whether it's a pickup or delivery; and *third*, for checking whether the selected restaurant is within the range of delivery service (in case of delivery order). 8. Final order will be placed.
Exceptions:	2a. If the credentials are invalid, the user will enter the login / register details again. 1. The selected item must be available. 2. If the order is out of range of the delivery service or the order is placed less than 30 min of the restaurant closing time, the order will be discarded. 6a. If provided payment details are incorrect then transaction will be cancelled.

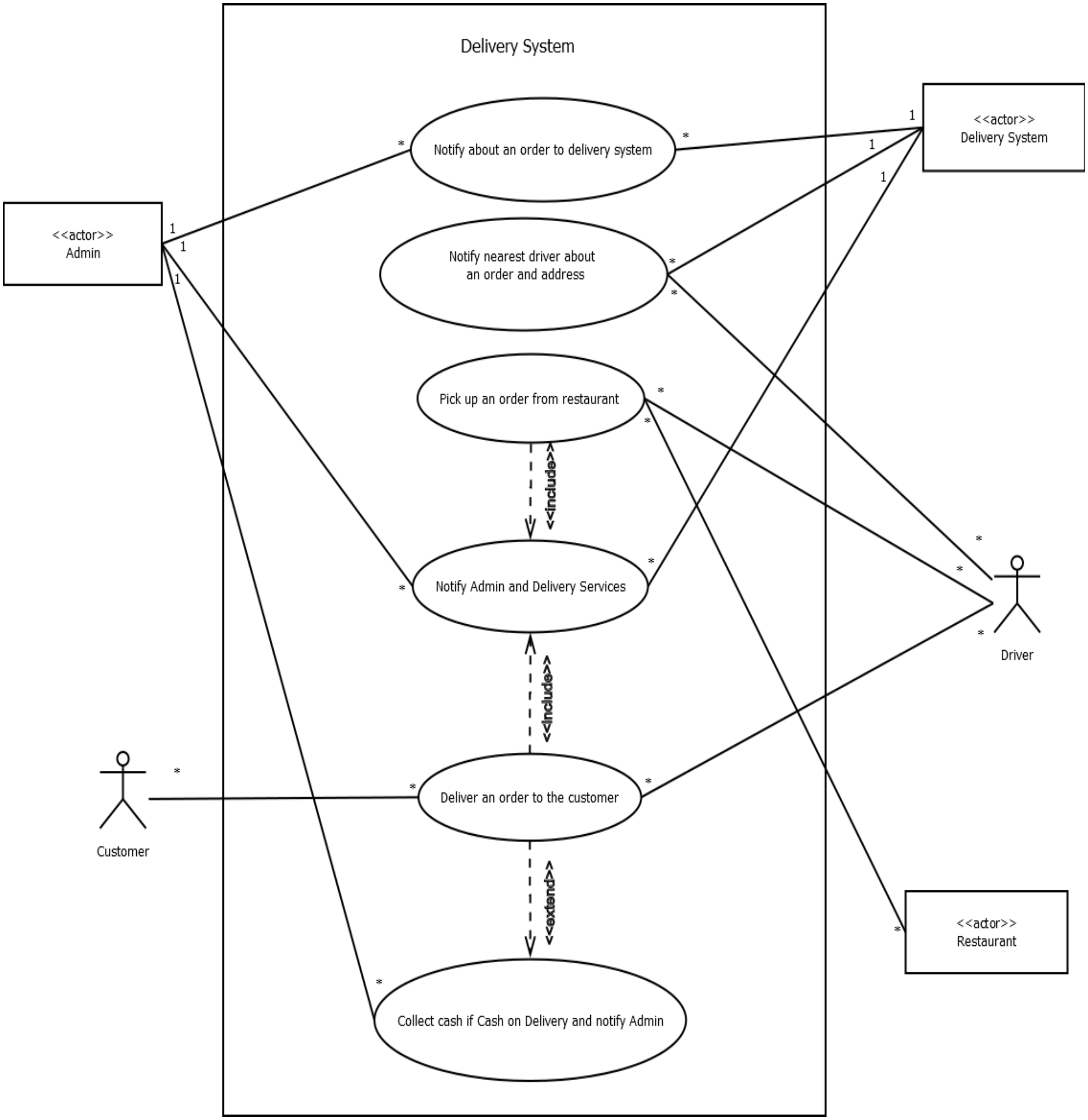
2. Restaurant Registration



Use Case Description for Restaurant Registration:

Case Name:	Restaurant Registration
ID:	UC_3
Actors:	Admin, Restaurant(client)
Description:	The Restaurant manager register his restaurant in this app so that it could be available to the all users(customers)
Trigger:	The restaurant should be available all the time and should be subscribed by the manager
Relationships:	Association: User, System, App
Normal flow of Events:	1. The restaurants will be displayed the based on rating and available items. 2. Restaurant will register on the app and submitted to the admin and credentials will be given Restaurant manager so that he can add offers, prices and update menu items on the daily basis. 3. The credentials will be validated by the system. 4. Restaurant will be given rating based on their service provided. 5. All payments to the restaurants were done once the order is placed by the customer. 6. All restaurants should have an option of delivering the products and ready for pickups. 7. It should be available for cash on delivery
Exceptions:	3. The menu items stocks should be available online all the time. 4. The distance from the customer to the restaurant should be managed well for a pickup or delivery

3. Delivery System

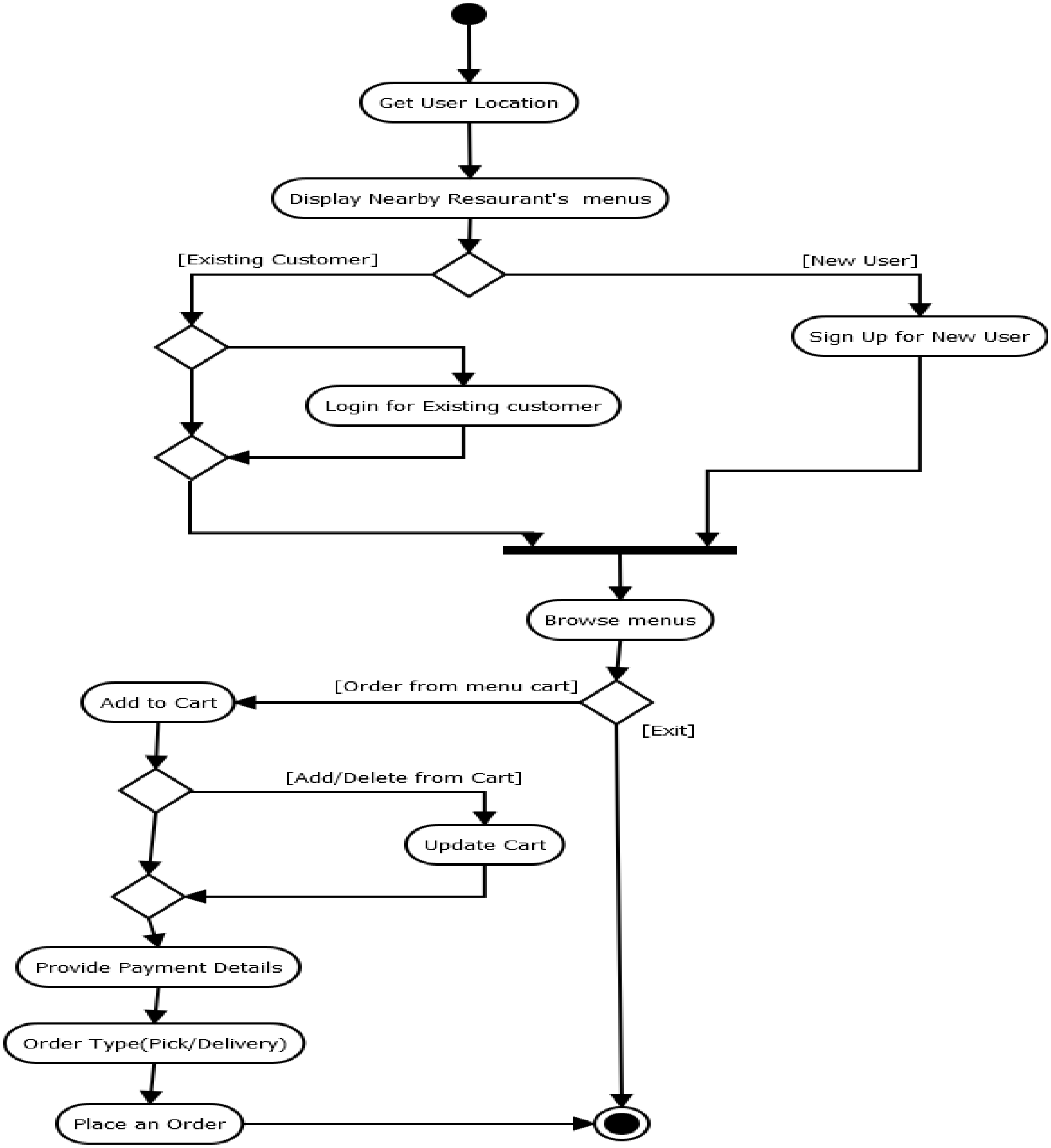

Delivery System
<<actor>>
Admin
<<actor>>
Delivery System
Notify about an order to delivery system
Notify nearest driver about
an order and address
Pick up an order from restaurant
<<include>>
Notify Admin and Delivery Services
<<include>>
Deliver an order to the customer
<<extend>>
Collect cash if Cash on Delivery and notify Admin
Customer
Driver
<<actor>>
Restaurant
1
1
1
1
1
1
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*

Use Case Description for Delivery of food:

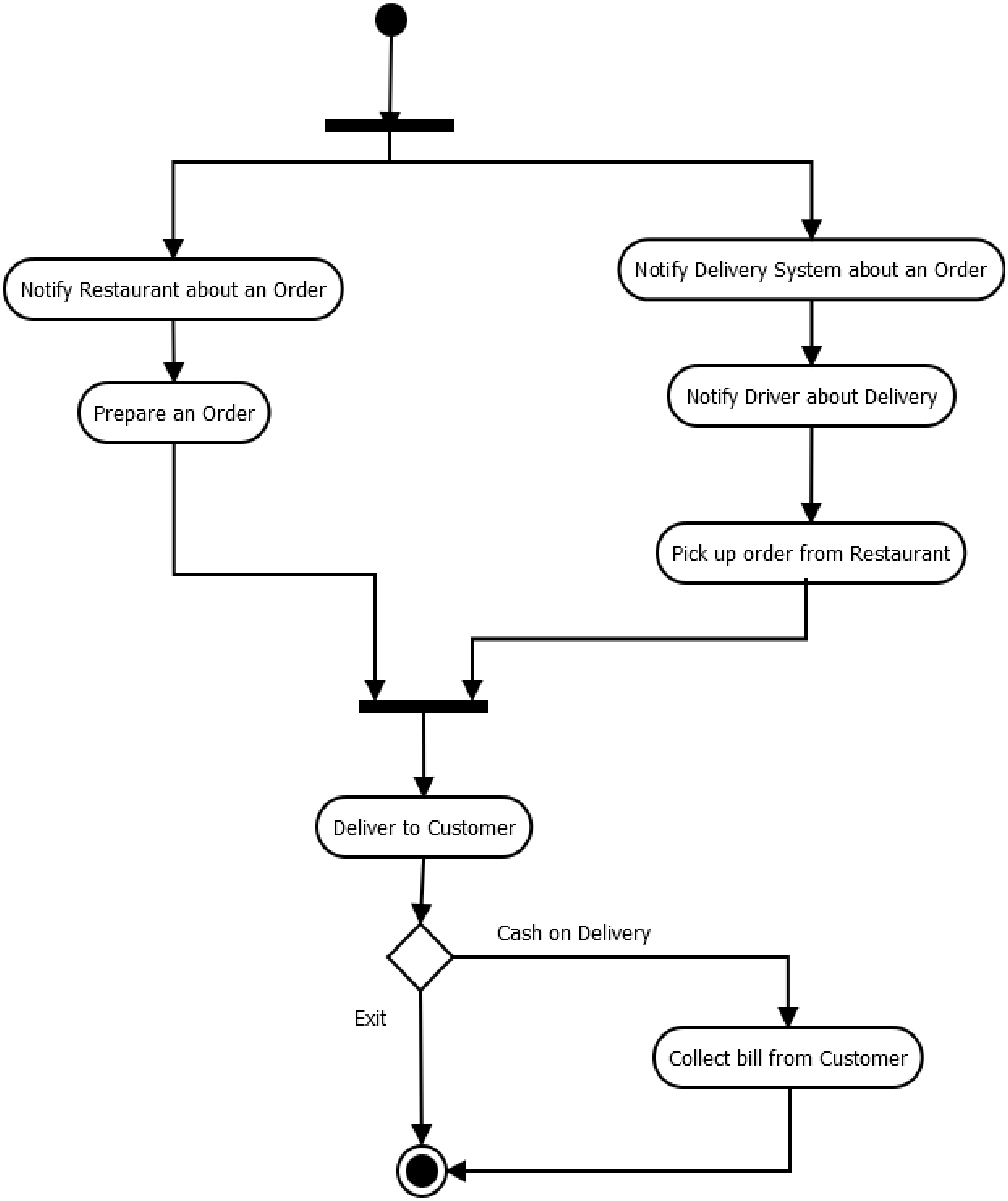
Use case Name	Delivery of food
ID	UC_3
Primary Actor	Delivery Services
Stakeholder and Investor	Customer: wants to order a food. Delivery services: wants to deliver a food.
Brief Description	This use case describe how delivery service deliver the food.
Trigger	While ordering food user has requested for home delivery.
Relationships	Association: Customer, Driver, Admin, Restaurant, Delivery System Include: Picking up an order and delivering a food to customer will include notifying Admin and Delivery System. Exclude: Cash on delivery will depend on customer’s choice. Generalization:
Normal flow of Events	1. Admin will notify the Delivery system about an order 2. Delivery system will notify the nearest driver about the order to pickup 3. Driver will pick up the order from specified restaurant and notification will be sent to Admin and Delivery System. 4. The Order will be delivered to customer and notification will be sent to Admin and Delivery System.
Sub flow	
Alternate/Exceptional Flow	4a. If cash on delivery, the Driver has to collect cash as indicated on the receipt.

3.4 Activity Diagram

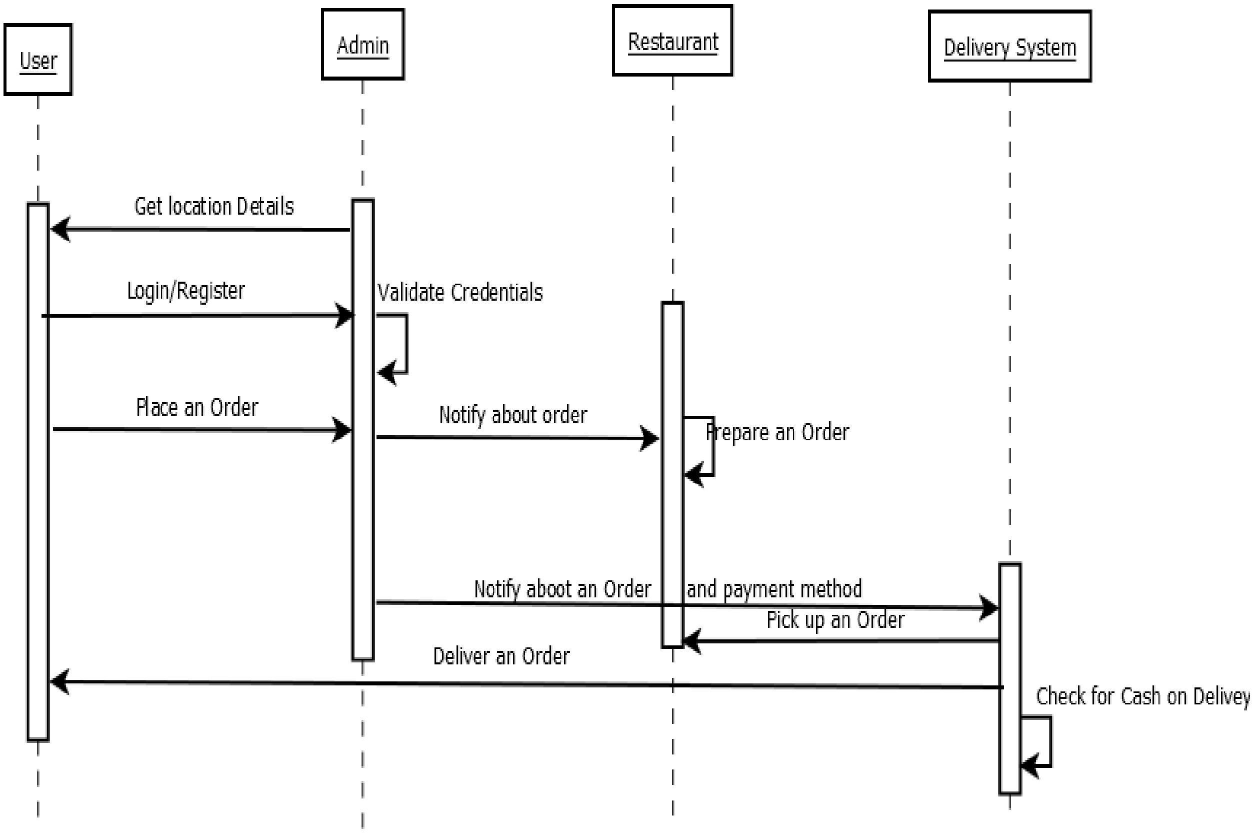
1. Activity Diagram for Placing an Order:



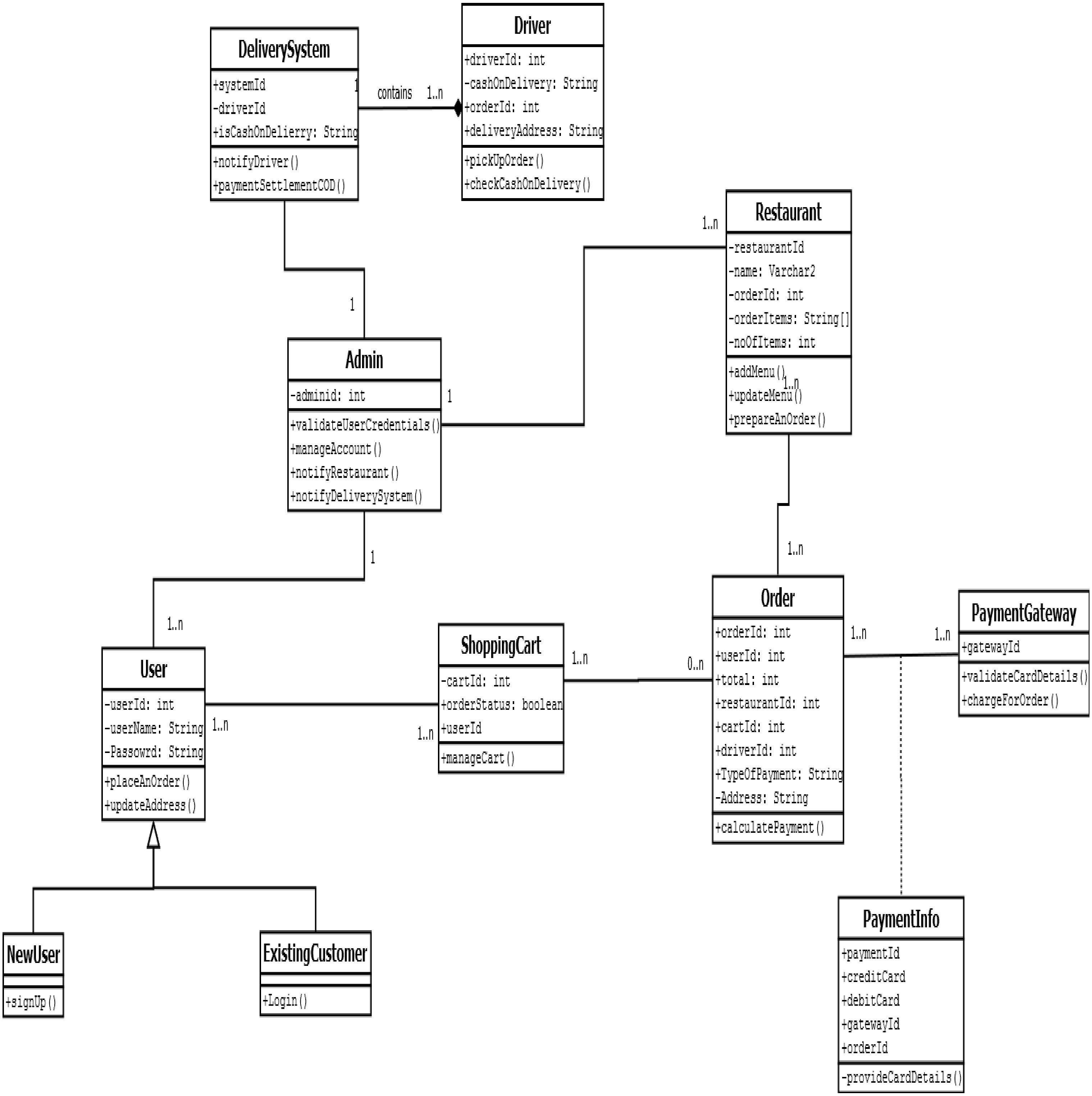
2. Activity Diagram for Restaurant and Delivery System:


Notify Restaurant about an Order
Notify Delivery System about an Order
Prepare an Order
Notify Driver about Delivery
Pick up order from Restaurant
Deliver to Customer
Cash on Delivery
Exit
Collect bill from Customer

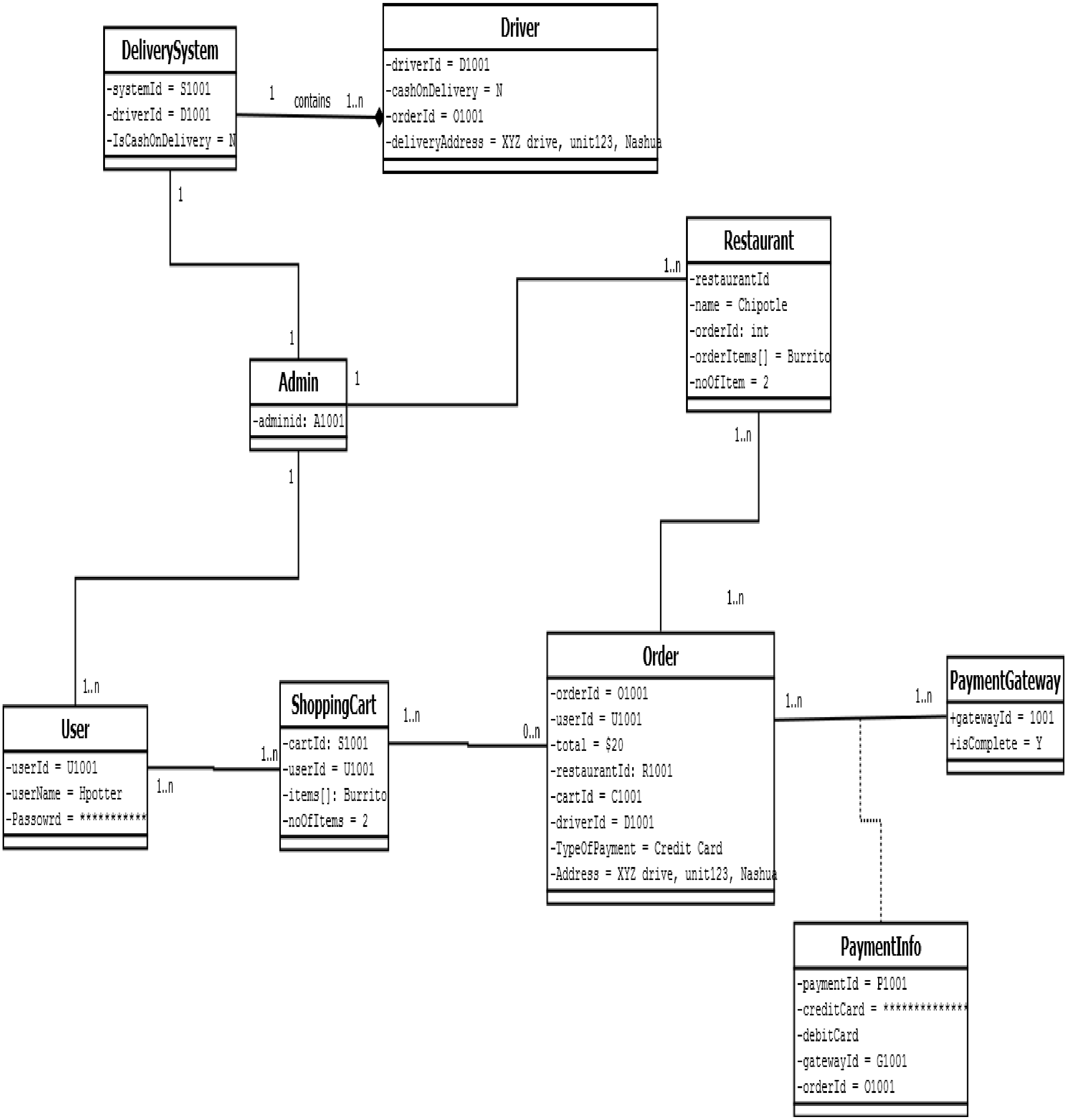
3.5 Sequence Diagram



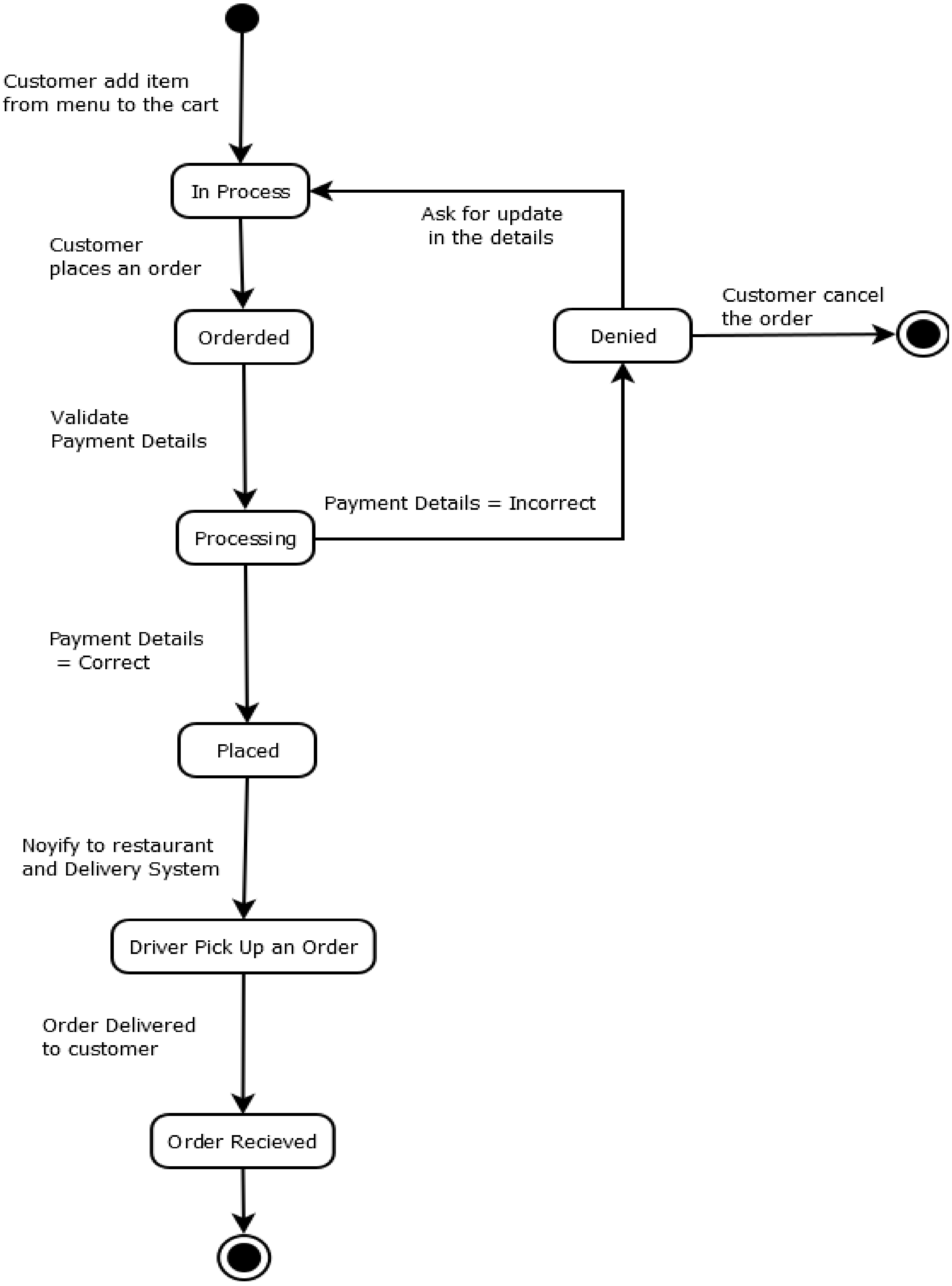
3.6 Class Diagram



3.7 Object Diagram



3.8 State Machine Diagram



4 Functional Test Plan

It would include all the functional test cases for the project.

Test Cases:

This section will provide an idea about how the functionality work after the software be ready to use. The test cases provide what are the valid or invalid data input for particular feature. Based in the inputs of the user the respective response will get.

4.1 Login test case for both User and Restaurant Panel

Action	**Input**	**Expected Output**	**Description**
Valid Login	Username, Password	Navigated to next screen	User Successfully Login into Food Order app.
Invalid Login	Username, Password	Alert box with message “Invalid Credentials” and ask for credentials again	Unauthorized user trying to login.

4.2 Test Case for Admin Panel

Action	**Input**	**Expected Output**	**Description**
Orders Approval	Orders approved from user	List of Orders in Queue which are out for Delivery	Orders approved
Update Ads Restaurant	Upload from computer and click on upload.	Display the Ads on UI	Ads Updated successfully.
Feedback	Select the Accessories	Display Feedback by user so far.	Feedback Successfully
Payments bifurcation	Select the Orders separate bills	Payment online through banking	Transactions successfully.

4.3 Test Case for User Panel

Action	Input	Expected Output	Description
Search Restaurants	Select restaurant from the list	Menu items of nearest restaurant to farther restaurant	Store Display
Confirm Order	Select the restaurant and confirm Order	Display list of orders of menu items in checkout	Confirm Order Successfully.
Add Contact and Address	Before Transaction the user enters Shipping and mobile details for delivery	No output	Delivery Address details
Give Feedback	Select the restaurant and choose the food from it and add comments for that respective order.	Update the list with users add comment or queries regarding the Admin comment.	Updated Comments successfully.
Pickup	Create order pickup from store	pickup order should be ready for pickup	Pickup drive successfully done.

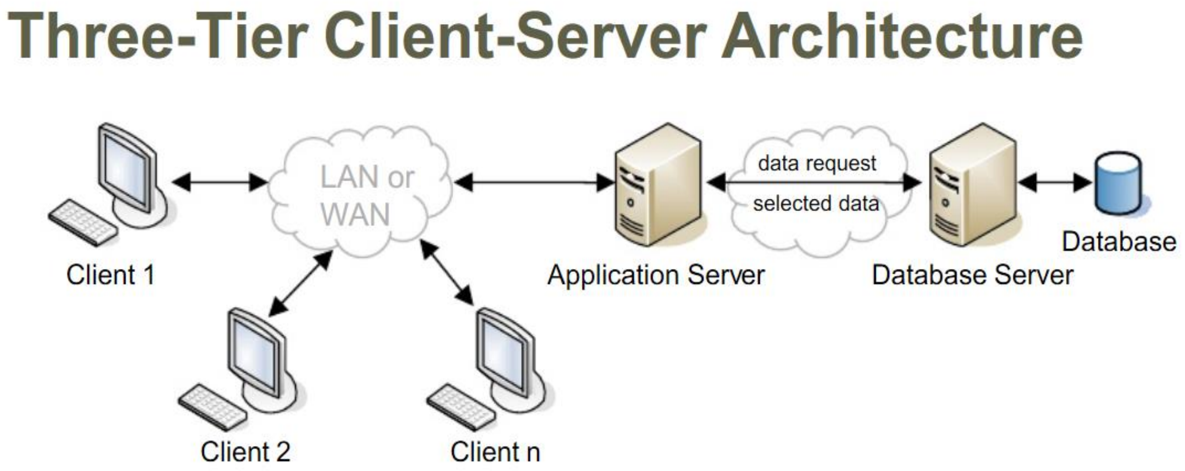
4.4 Test Case for Restaurant Panel

Action	Input	Expected Output	Description
Add Restaurant	Available food items in store	Display list of item available	Food available.
Delivery or Pickup	Selected the respective order check either pickup or delivery	There no specific output for this action.	Trigger a mail to customer and User regarding about “food is ready”
Give Feedback	Select the food and choose the option comments for that respective issue.	Positive feedback or negative feedback	Updated Comments successfully.
Bank transaction	Admin do the Transaction transfer the amount of cash individually for an order.	No output	Transaction will be done successfully.

5 System Design Specifications

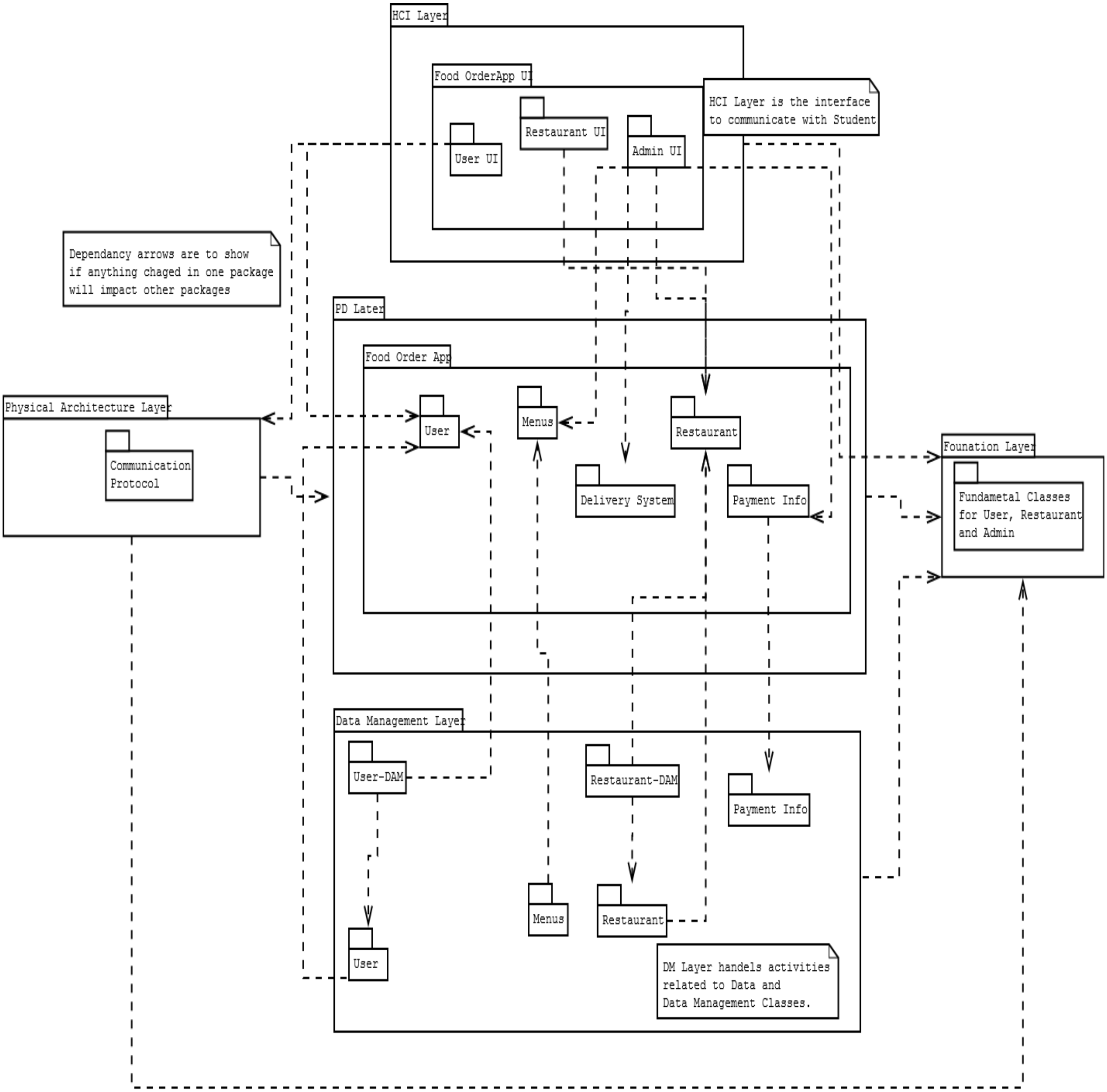
5.1 System Physical Architectural

The Food Order App will need Three Tier Client-Server Architecture, as, it is an online mobile application.



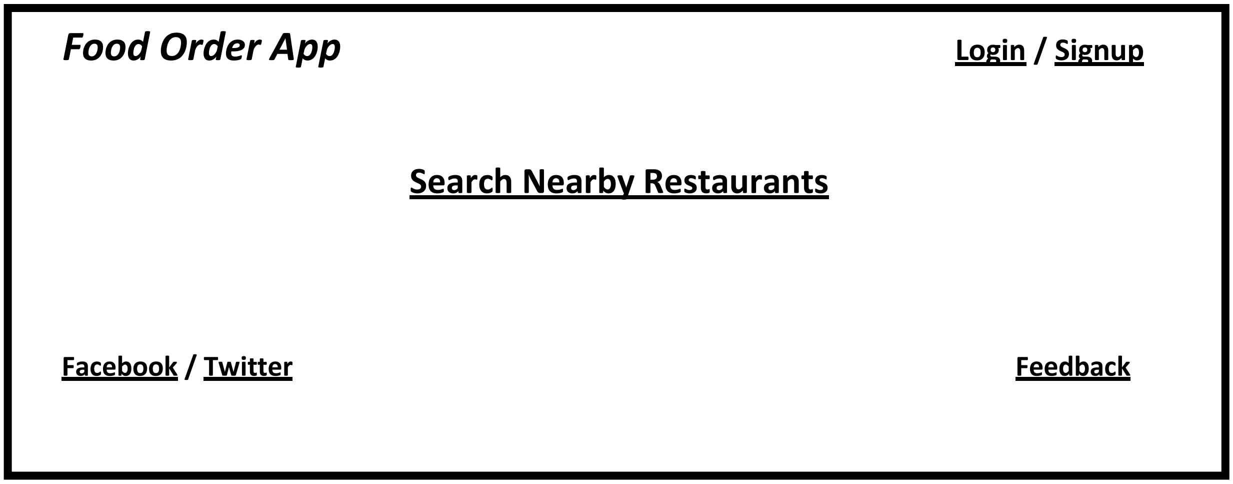
The above diagram describes how the Three Tier Client-Server Architecture will work. In the case of Food Order App clients will be the USER's who installed app on the mobile phone and RESTAURANT's who are registered for the app. Application Server will take care of the USER INTERFACE and Validations. Database Server will store all the data related to the application and will communicate with Application Server for validating all USER's.

5.2 Package Diagram

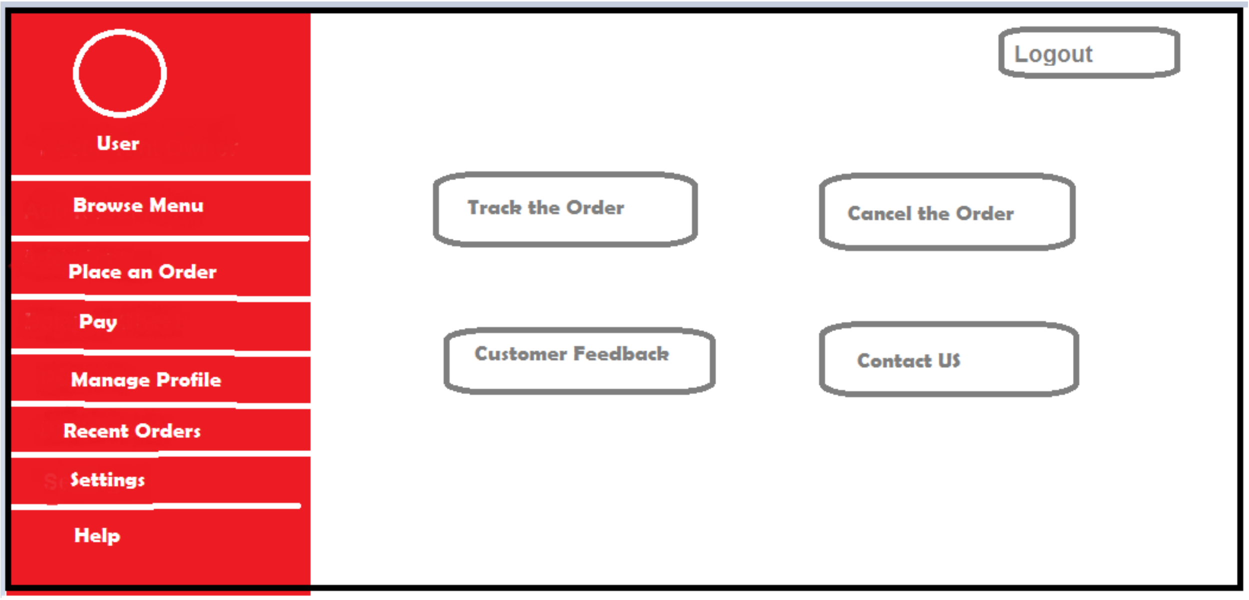

HCI Layer
Food OrderApp UI
User UI
Restaurant UI
Admin UI
HCI Layer is the interface to communicate with Student
Dependancy arrows are to show if anything chaged in one package will impact other packages
PD Later
Food Order App
User
Menus
Restaurant
Delivery System
Payment Info
Physical Architecture Layer
Communication Protocol
Founation Layer
Fundametal Classes for User, Restaurant and Admin
Data Management Layer
User-DAM
Restaurant-DAM
Payment Info
Menus
Restaurant
User
DM Layer handels activities related to Data and Data Management Classes.

5.3 UI Design

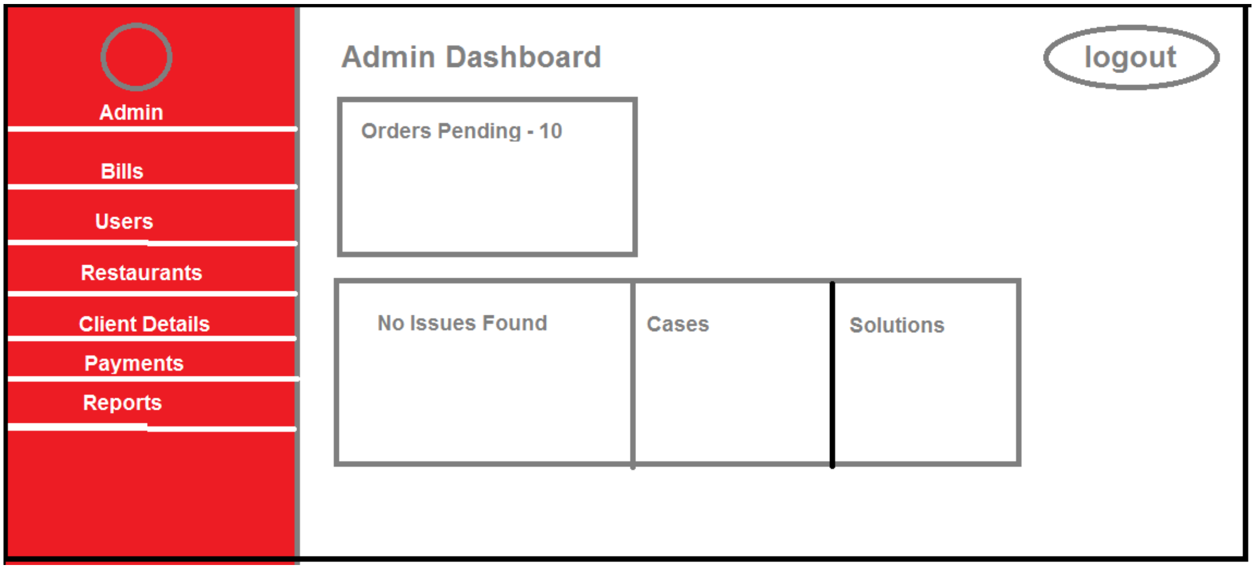
Login/Register UI



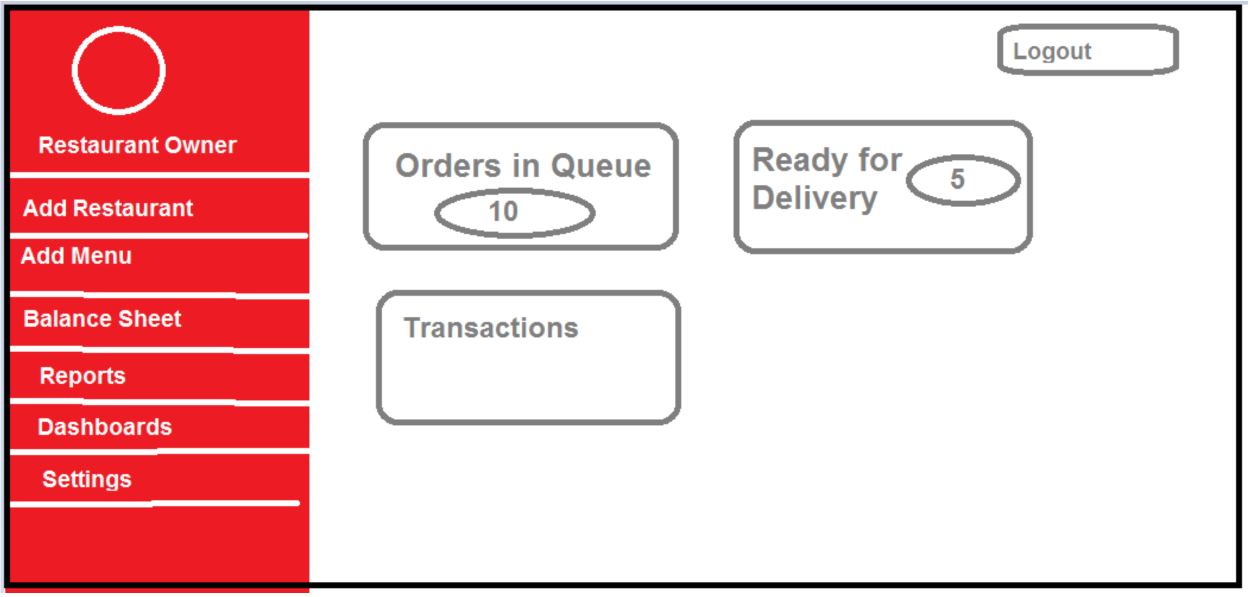
User's UI



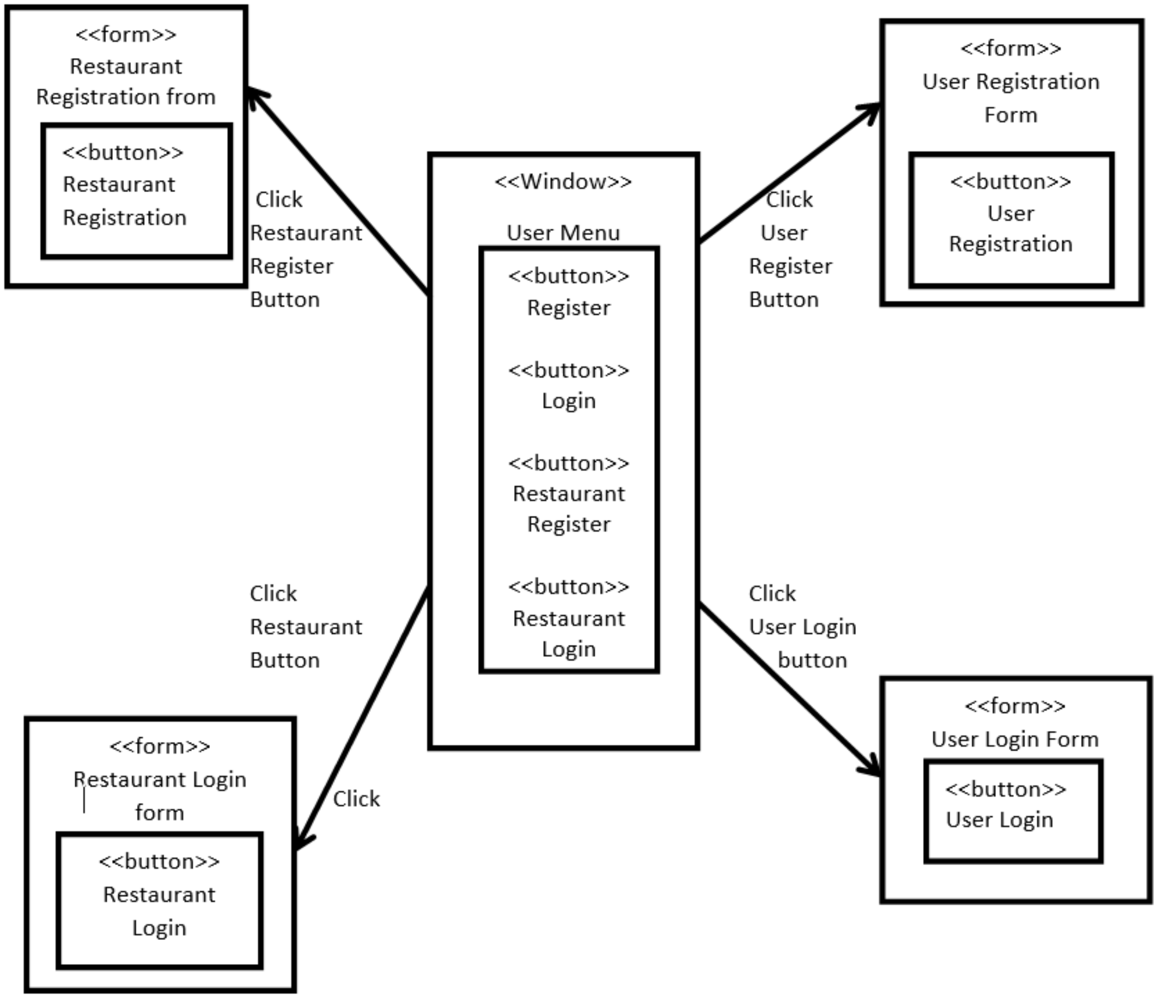
Admin UI



Restaurant UI



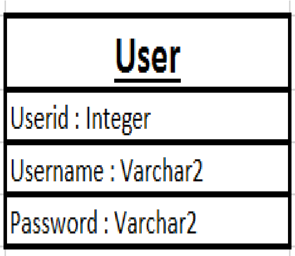
5.4 Window Navigation Diagram



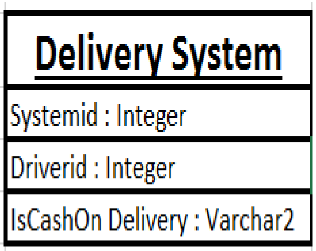
6 Database Structure

6.1 Table Structure

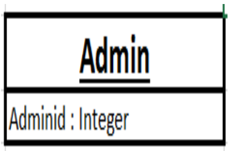
Below are some important table structures for the project:



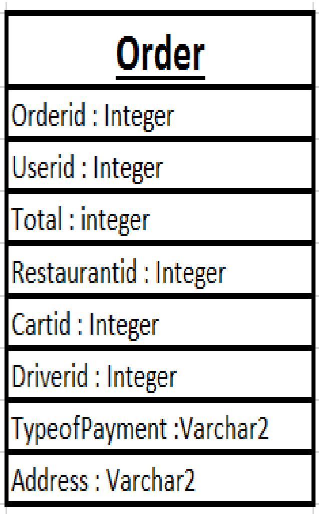
User
Userid : Integer
Username : Varchar2
Password : Varchar2



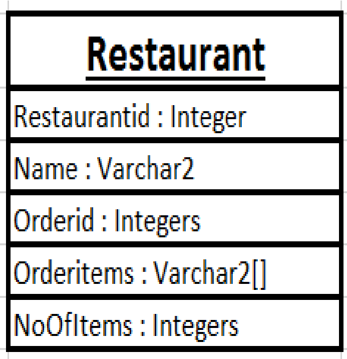
Delivery System
Systemid : Integer
Driverid : Integer
IsCashOn Delivery : Varchar2



Admin
Adminid : Integer

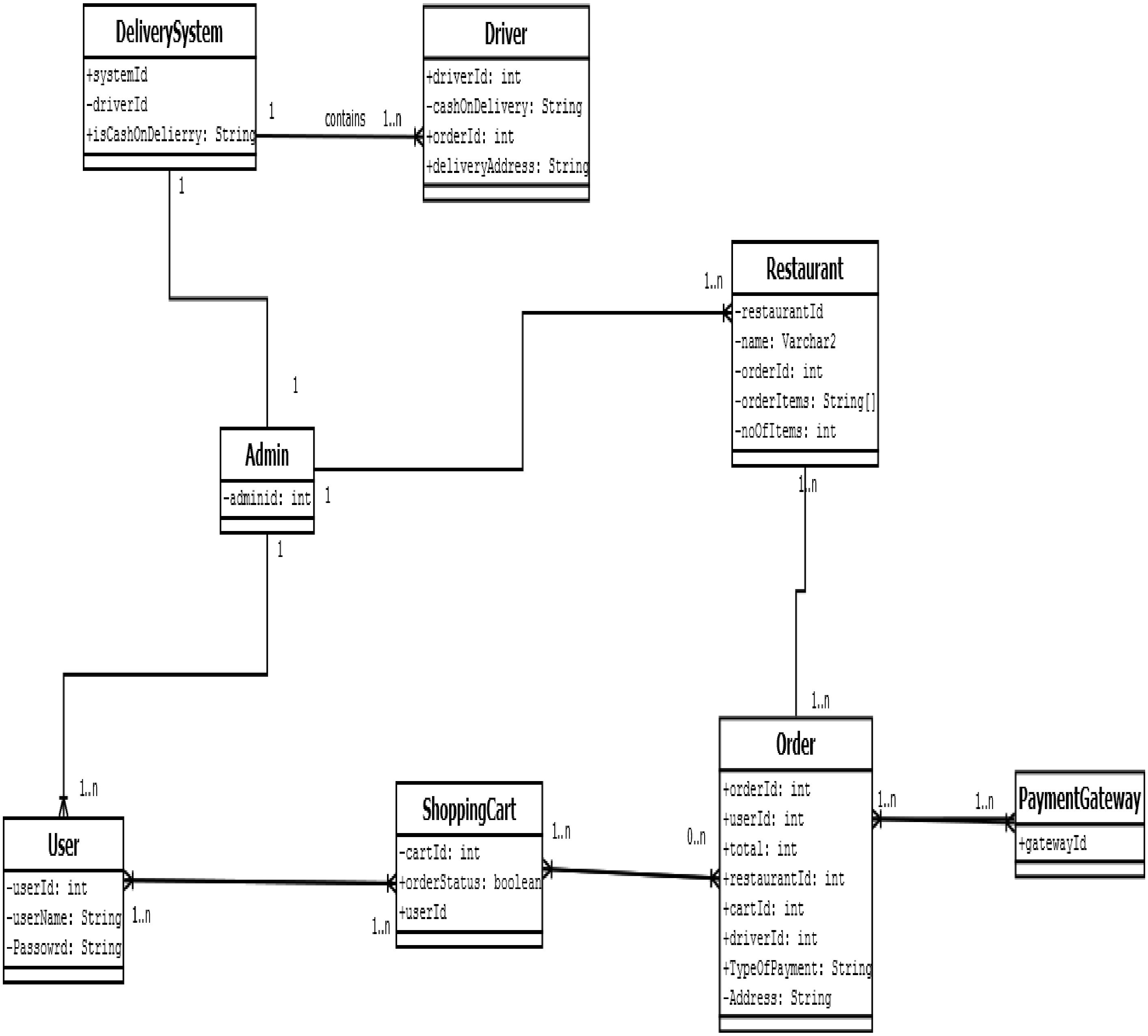


Order
Orderid : Integer
Userid : Integer
Total : integer
Restaurantid : Integer
Cartid : Integer
Driverid : Integer
TypeofPayment :Varchar2
Address : Varchar2



Restaurant
Restaurantid : Integer
Name : Varchar2
Orderid : Integers
Orderitems : Varchar2[]
NoOfItems : Integers

6.2 ER Diagram



7 System Integration Test Plan

Below are the test cases which should be covered into system integration test plan, as they indicate how the system behaves when all components are put together.

Test case 1: User request for the order summary

Expected Output: App should display order summary by retrieving it from database

Test Case 2: User places an Order

Expected Output: Notification should be sent to Restaurant and Delivery System with all details

Test Case 3: User places an Order, Email notification test

Expected Output: Once the order has been placed, user should get email from the system.

8 Issues for future studies

The following project can be enhanced by adding below functionalities:

1. GPS system can be added into system, so the system can directly understand location of the user.
2. User should able to see the order status e.g. Order ready, Picked up by Driver etc.
3. The order data collected from the app can be used to analyze food preferences of an area.

9 Conclusion

Food Order App will give convenient approach for customer to browse an order food and for Restaurant to grow their market.

10 References

1. For project idea: https://www.foodler.com/
2. For System Physical Architectural

https://dbinfoworld.wordpress.com/2014/11/09/different-types-of-database-architecture/