NOAKHALI SCIENCE & TECHNOLOGY UNIVERSITY

Noakhali -3814

Lab Report On

Software requirements specification (SRS) for an online food ordering system.

Course Code: ICE-4101

Submitted By

Naser Mahmud
Roll: ASH1811070M

Department of ICE

Session: 2017-18(Re-add)

Submitted To

Md. Mahbubul Alam

Lecturer

Department of ICE

NSTU

Date of Submission:07-08-2023

Name of the Experiment: Software requirements specification (SRS) for an online food ordering system.

Introduction: This lab report describes the software requirements specification (SRS) for an online food ordering system. The system will allow customers to order food from a restaurant online, and for the restaurant to manage their orders and deliveries.

Purpose : The purpose of this SRS is to outline both the functional and non-functional requirements of the subject Restaurant food ordering system. 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 document should act as a foundation for efficient and well-managed project completion and further serve as an accurate reference in the future.

Scope: The system will be used by customers to view the menu of a restaurant, select items to order, and pay for their order. The system will also be used by the restaurant to manage their orders, including assigning orders to delivery drivers, tracking the status of orders, and sending notifications to customers.

Background and Related Work : This Case study looks at the problem of setting up a fast food restaurant. In existing system there are few problems:

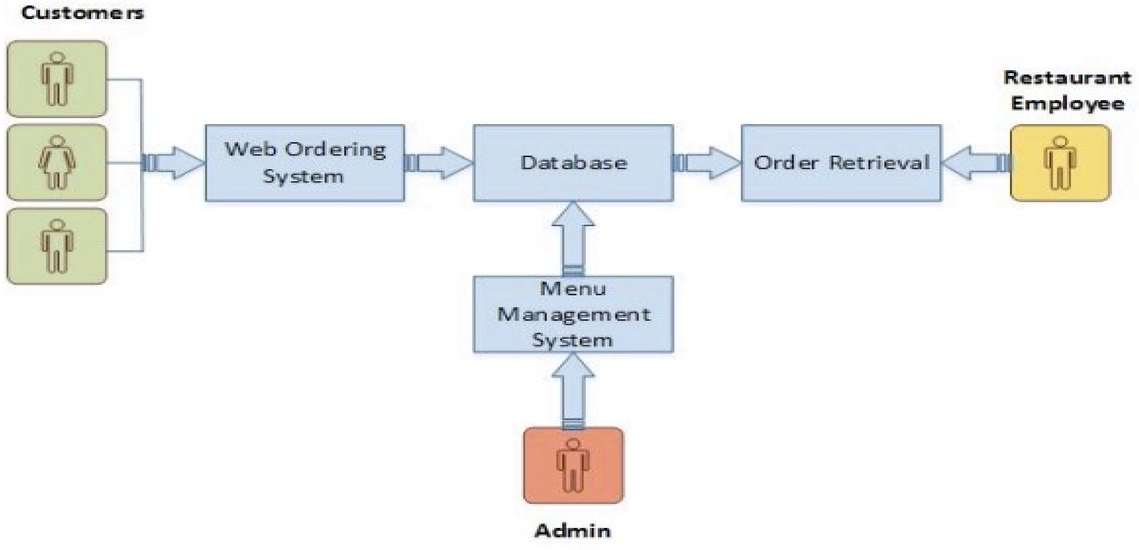
- For placing any orders customers have to visit hotels or restaurants to know about food items and then place order and pay. In this method time and manual work is required.
- While placing an order over the phone, customer lacks the physical copy of the menu item, lack of visual confirmation that the order was placed correctly.
- Every restaurant needs certain employees to take the order over phone or in-person, to offer a rich dining experience and process the payment. In today’s market, labor rates are increasing day by day making it difficult to find employees when needed.

Hence, to solve this issue, what I propose is an “Online Food Order System, originally designed for small scale business like College Cafeterias, Fast Food restaurant or Take-Out, but this system is just as applicable in any food delivery industry.

The main advantage of my system is that it greatly simplifies the ordering process for both the customer and the restaurant and also greatly lightens the load on the restaurant’s end, as the entire process of taking orders is automated. Anticipated Benefits are:

- 1. This will minimize the number of employees at the back of the counter.
- 2. The system will help to reduce labor cost involved.
- 3. The system will be less probable to make mistake, since it’s a machine.

System Model:



The structure of the system can be divided into 3 main logical components:

(A)Web Ordering System Module This module provides the functionality for customers to place their order and supply necessary details. Users of the system, namely restaurant customers, must be provided the following functionality:

- ◆ Create an account.
- ◆ Manage their account.
- ◆ Log in to the system.
- ◆ Navigate the restaurant's menu.
- ◆ Select an item from the menu.
- ◆ Add an item to their current order

(B)Menu Management System Module: This module provides functionality for the power user-Administrator only. It will not be available to any other users of the system like Restaurant Employees or Customers. Using a graphical interface, it will allow an Admin to manage the menu that is displayed to users of the web ordering system:

- ◆ Add/update/delete food category to/from the menu.
- ◆ Add /update/delete food item to/from the menu.
- ◆ Update price for a given food item.

(C)Order Retrieval System Module This is the most simplest module out of all 3 modules. It is designed to be used only by restaurant employees, and provides the following functions:

- ◆ Retrieve new orders from the database.
- ◆ Display the orders in an easily readable, graphical way

Functional Requirements: The system must meet the following functional requirements:

■Customers must be able to view the menu of a restaurant.
■Customers must be able to select items to order.
■Customers must be able to pay for their order.
■The restaurant must be able to manage their orders.
■The restaurant must be able to assign orders to delivery drivers.
■The restaurant must be able to track the status of orders.

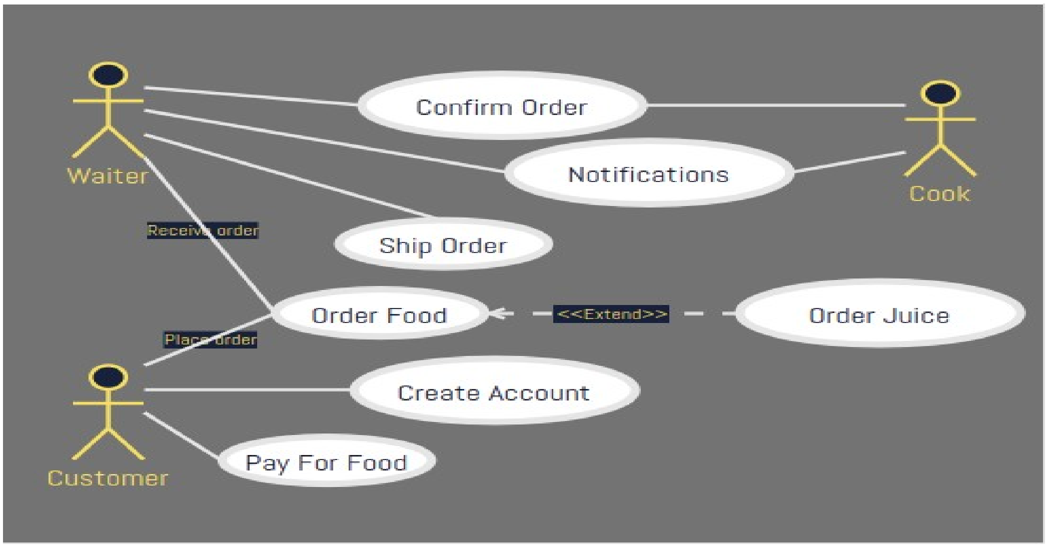
■The restaurant must be able to send notifications to customers.

Non-Functional Requirements: The system must meet the following non-functional requirements:
■The system must be available 24/7.
■The system must be able to handle a large number of concurrent users.
■The system must be secure from unauthorized access.
■The system must be reliable and consistent.
■The system must be easy to use.

Assumptions and Constraints: The following assumptions and constraints apply to the system:
■The system will be used by customers with a variety of devices, including computers, smartphones, and tablets.
■The system will be used by restaurants of all sizes.
■The system will be used in a variety of countries.

Use Cases: The following use cases describe the functionality of the system:
◆Customer Order: A customer orders food from a restaurant.
◆Restaurant Order Management: A restaurant manages their orders.
◆Delivery Driver Assignment: A restaurant assigns orders to delivery drivers.
◆Order Status Tracking: A restaurant tracks the status of orders.
◆Customer Notification: A restaurant sends notifications to customers.

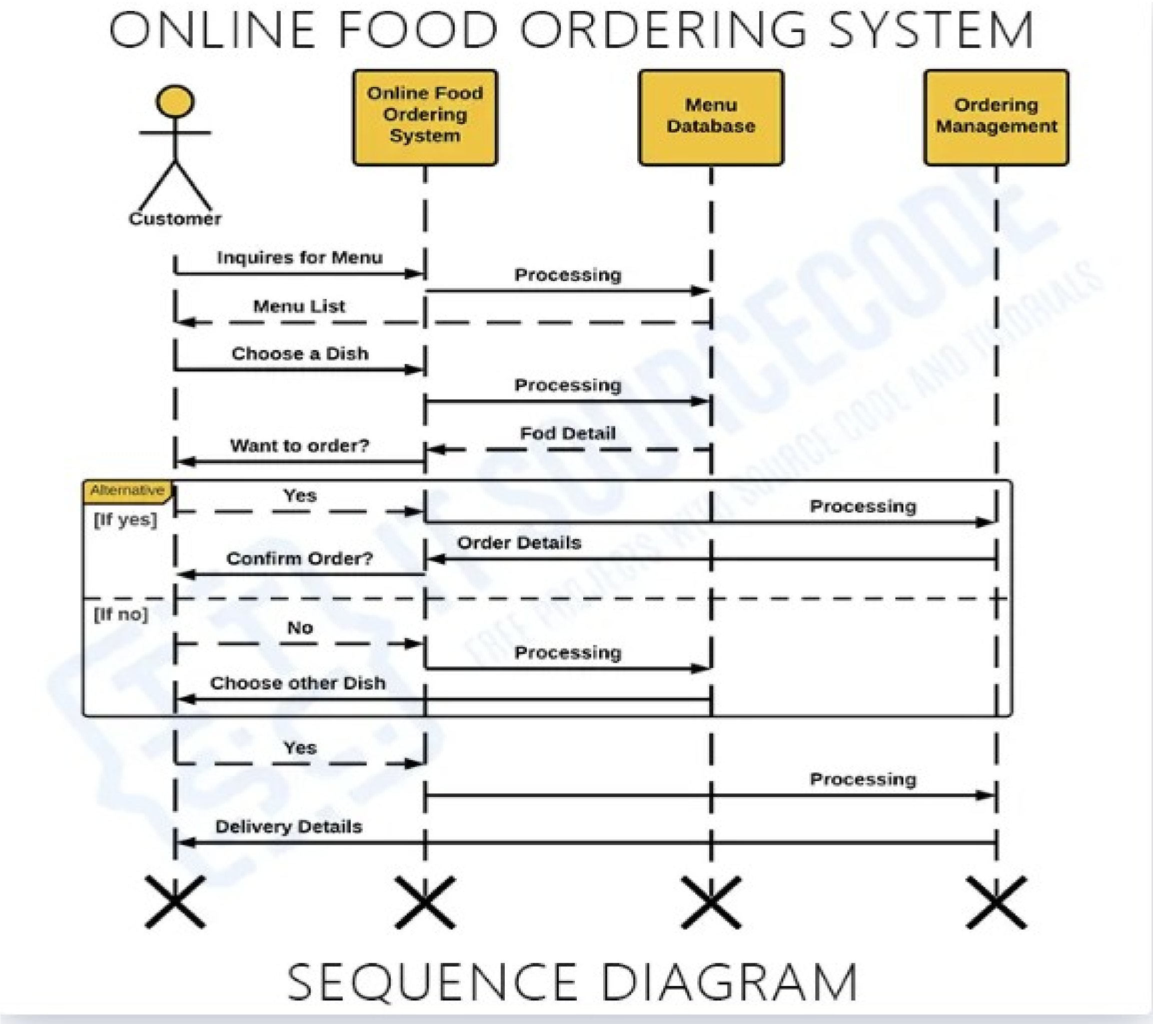
Diagrams: The following diagrams are included in the SRS:
Use case diagram: This diagram shows the different use cases of the system, as well as the actors involved in each use case.



Use case diagram for online food ordering system

Sequence diagram: This diagram shows the sequence of interactions between the system and the actors in a specific use case.
Here, I will be showing you the *Online Food Ordering System Sequence Diagram*. This design will enlighten you on how should the system or the actor approach each other.
This will also teach you on how would you develop the system to achieve its desired behavior. In addition to that, the design shows the detailed illustration of events sequenced and happens in Food Ordering System.
This designed sequence diagram is able to show programmers and readers about the sequence of messages between the actor and the objects.

As you can see through the illustration, the conditions and interactions are emphasized. These interactions are essential for the Online Food Ordering System development.
The series of messages are shown and labeled to guide you in building the System. You can modify the design if you have more ideas. You can also add more features to this design and use it as your project blueprint.
You'll be able to understand and educate yourself on how the Food Ordering System works by creating a sequence diagram. Because it determines the needed objects, actors and messages and their interactions.



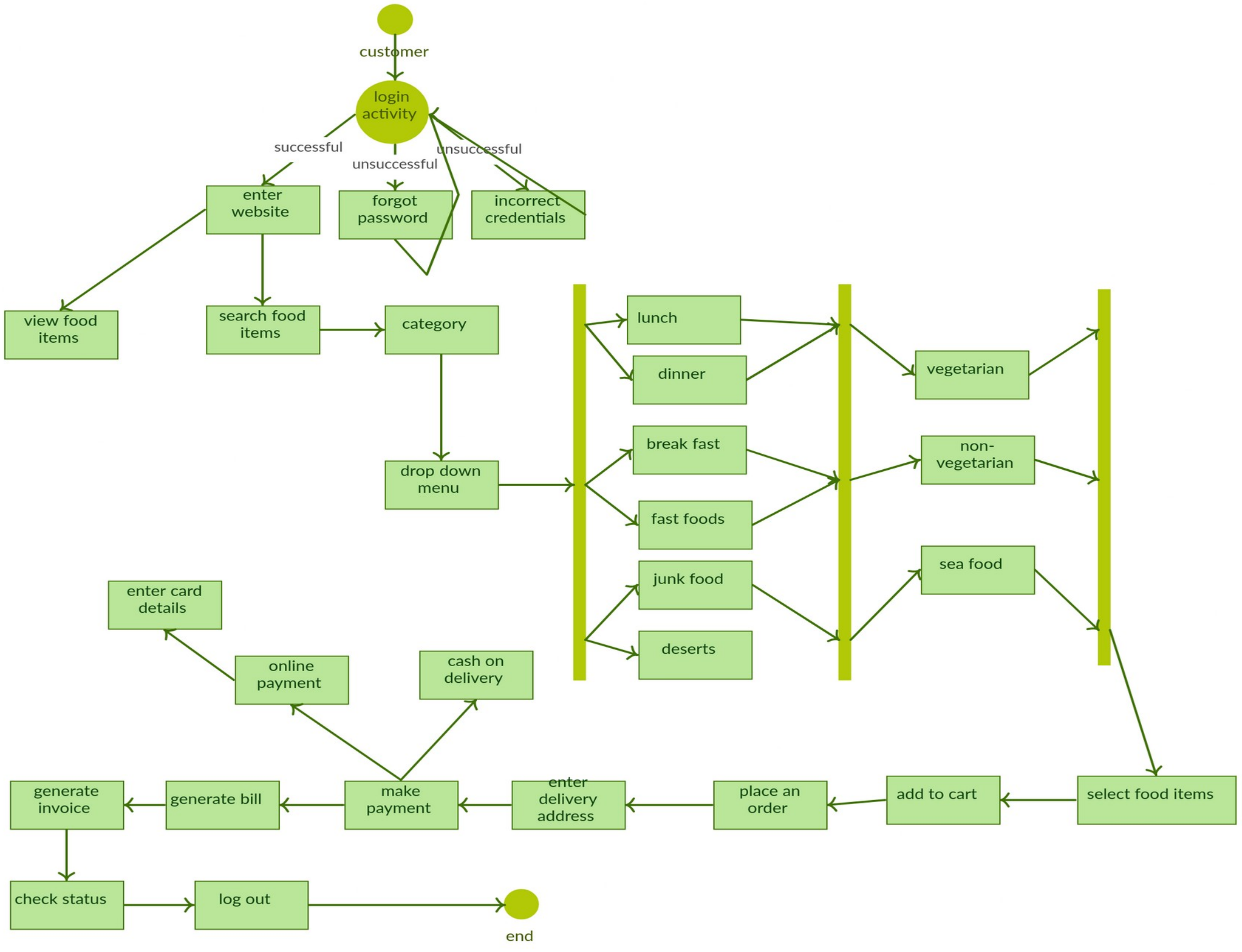
State machine diagram:

The diagram shows the different states that an order can be in, as well as the transitions between states. The diagram starts in the * state, which represents the initial state. From the initial state, the user can select a restaurant. Once a restaurant has been selected, the user can view the menu. The user can then select items from the menu and add them to the cart. Once the user is satisfied with their order, they can review it. If the user is happy with the order, they can pay for it. Once the order has been paid for, it is placed. The order is then confirmed, and the system returns to the initial state.

The diagram also shows some of the events that can trigger transitions between states. For example, the Select Restaurant event triggers the transition from the initial state to the Select Restaurant state. The View Menu event triggers the transition from the Select Restaurant state to the View Menu state. The Add to Cart event triggers the transition from the View Menu state to the Add to Cart state. And so on.

The state machine diagram is a useful tool for visualizing the different states that an order can be in and the transitions between states. The diagram can be used to help developers understand the flow of control in the system and to identify potential problems.

This diagram shows the different states that the system can be in, as well as the transitions between states.



State machine diagram for online food ordering system

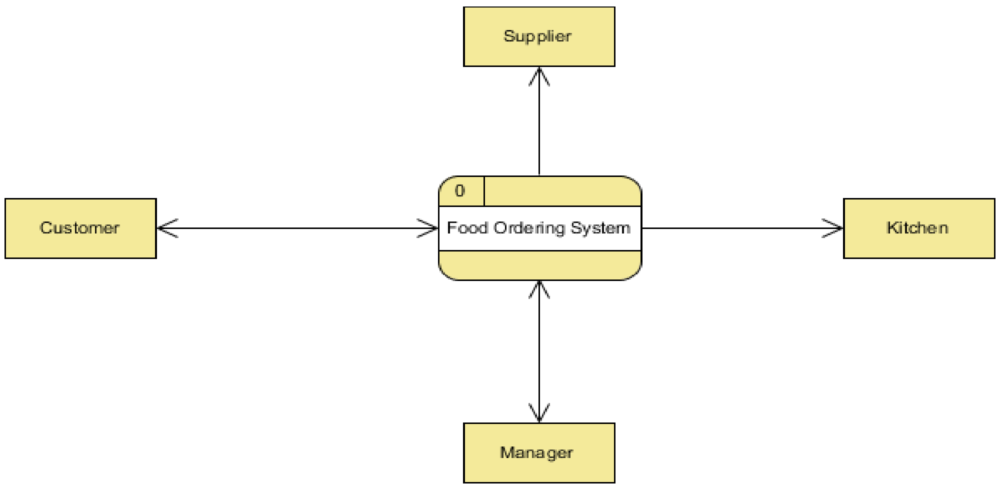
Data flow diagram: Data Flow Diagram (DFD) provides a visual representation of the flow of information (i.e. data) within a system. By drawing a Data Flow Diagram, you can tell the information provided by and delivered to someone who takes part in system processes, the information needed to complete the processes and the information needed to be stored and accessed. This article describes and explains the Data Flow Diagram (DFD) by using a food ordering system as an example.

Context DFD

A context diagram is a data flow diagram that only shows the top level, otherwise known as Level 0. At this level, there is only one visible process node that represents the functions of a complete system in regards to how it interacts with external entities. Some of the benefits of a Context Diagram are:

- Shows the overview of the boundaries of a system
- No technical knowledge is required to understand with the simple notation
- Simple to draw, amend and elaborate as its limited notation

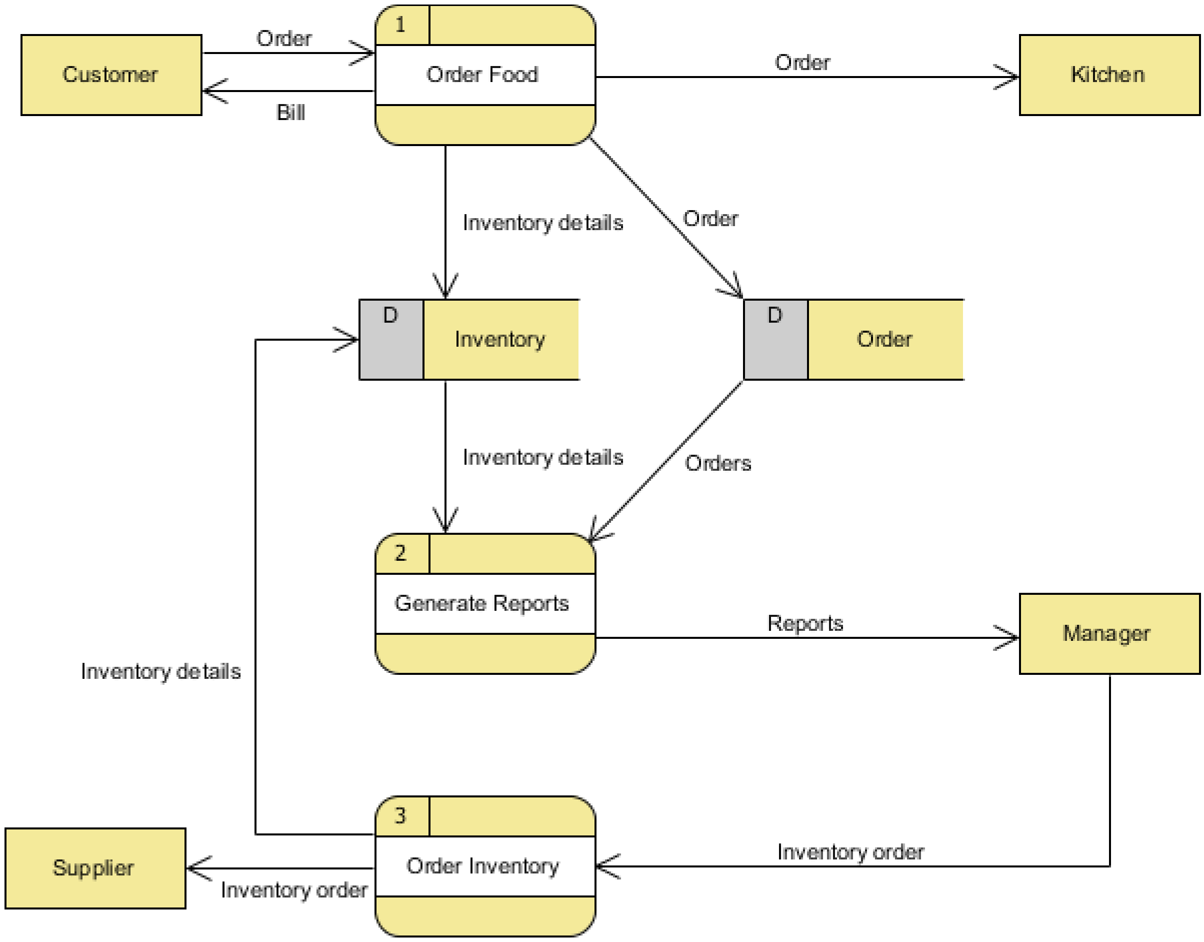
The figure below shows a context Data Flow Diagram that is drawn for a Food Ordering System. It contains a process (shape) that represents the system to model, in this case, the "Food Ordering System". It also shows the participants who will interact with the system, called the external entities. In this example, theSupplier,Kitchen,Manager, and Customer are the entities who will interact with the system. In between the process and the external entities, there is data flow (connectors) that indicate the existence of information exchange between the entities and the system.



Context DFD is the entrance of a data flow model. It contains one and only one process and does not show any data store.

Level 1 DFD

The figure below shows the level 1 DFD, which is the decomposition (i.e. break down) of the Food Ordering System process shown in the context DFD. Read through the diagram and then we will introduce some of the key concepts based on this diagram.



Data flow diagram for online food ordering system

Conclusion: The SRS for the online food ordering system has been described in this lab report. The system will meet the needs of customers and restaurants, and it will be easy to use and secure. The diagrams included in the SRS will help developers understand the requirements of the system and implement it correctly.