Ryerson University

Department of Electrical, Computer, & Biomedical Engineering
Faculty of Engineering & Architectural Science

Course Title:	Software REQs Analysis SPEC
Course Number:	COE691
Semester/Year (e.g.F2016)	W2024

Instructor:	Rasha Kashef

Assignment/Lab Number:	4
Assignment/Lab Title:	Context. DFD, ER and State Dragrams

Submission Date:	2024/03/25
Due Date:	2024/03/25

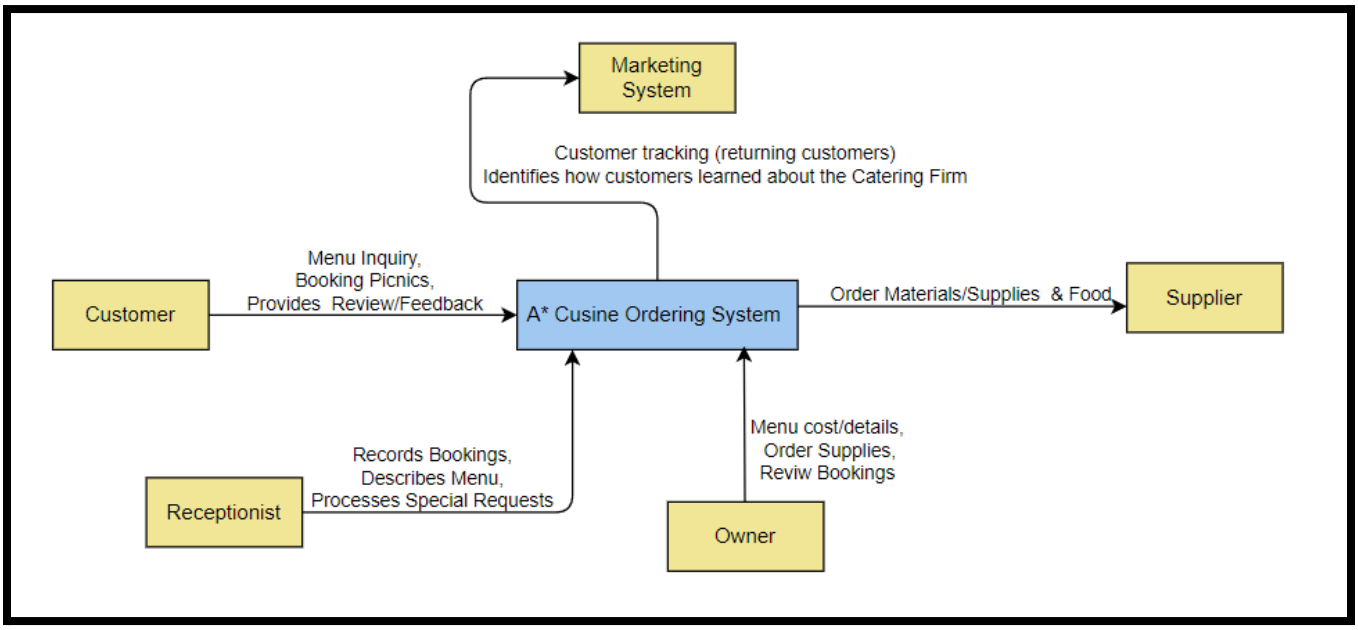
Student LAST Name	Student FIRST Name	Student Number	Section	Signature*
Sarim	Shahwar	501109286	02	SS

*By signing above you attest that you have contributed to this written lab report and confirm that all work you have contributed to this lab report is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at: http://www.ryerson.ca/senate/current/pol60.

Part I: Context and DFD Diagrams

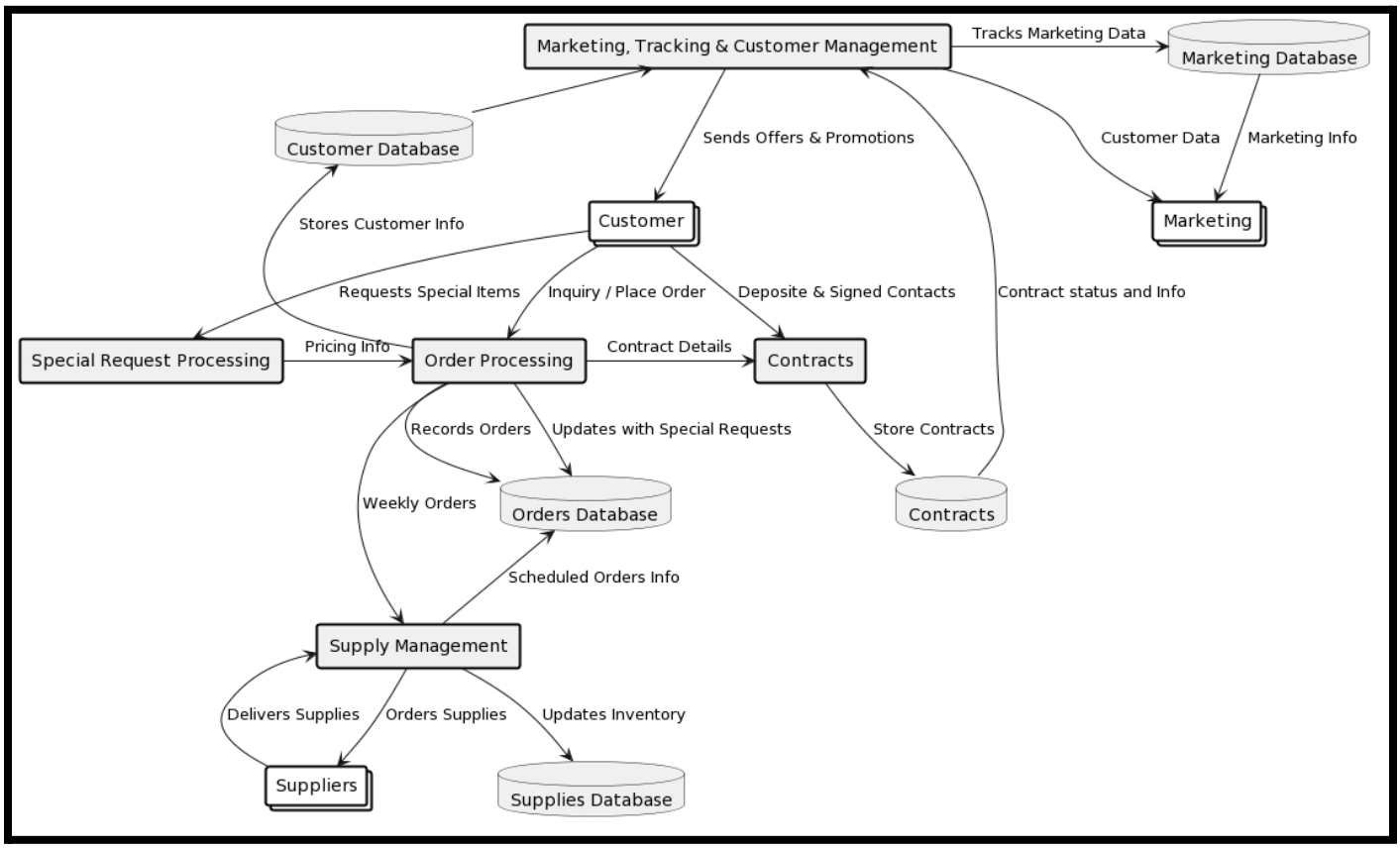
Case Study: A* Cuisine Firm

Context Diagram:



The context diagram shows the critical parts of the A* Cuisine Ordering System for a catering company. It clearly displays how the system interacts with different outside groups. The main system acts as the center for data and communication between the customers, receptionist, owner, and supplier, as well as an external marketing system. Customers use the system to check menus, make reservations, and leave feedback. The receptionist utilizes the system to manage bookings, provide menu details, and handle special requests. The owner accesses the system to view menu costs, place supply orders, and review bookings, reflecting their management and oversight role. Suppliers receive orders through the system, which is crucial for obtaining the necessary materials and food. The marketing system sends and gets details about how clients were obtained and tracked. It handles data exchanges related to customer acquisition sources. This high-level diagram makes complex operations look simple. Its visual simplicity helps possible stakeholders grasp the big picture of the system.

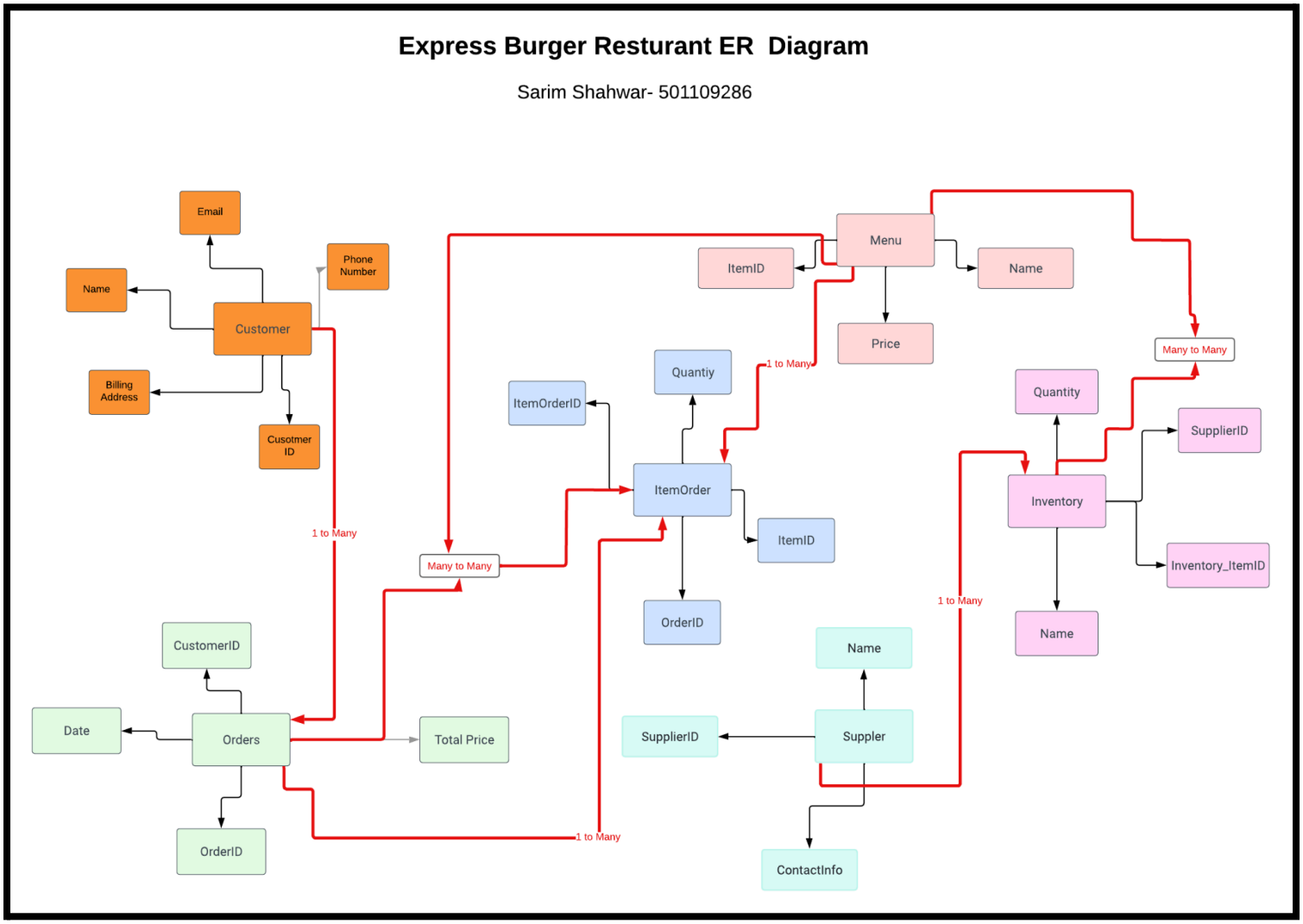
DFD Diagram:



The Data Flow Diagram (DFD) illustrates the flow of information between various components of a company's system, which include marketing, order processing, and supply management. Central to this system is the customer, who interacts with the system by making inquiries, placing orders, and receiving tailored marketing communications such as offers and promotions. Customer-related information is systematically stored and managed in a Customer Database, which is crucial for both the marketing department and order processing. The Marketing department utilizes a dedicated Marketing Database to track and manage marketing data and to strategize customer engagement. Meanwhile, contracts are a crucial element in formalizing the terms of service, and their details are stored and tracked.
The order processing function is a bridge between customer requests, which includes special item requests, and the fulfillment of these orders, updating the Orders Database with all transaction details. Special Request Processing is tasked with evaluating and pricing these unique customer requests, feeding this information back into the order processing flow. On the supply side, Supply Management coordinates with Suppliers to maintain an inventory of necessary items, all tracked within a Supplies Database. This ensures a seamless operation from the point of customer interaction through order fulfillment to the management of supplies, allowing the company to operate efficiently and responsively to customer needs.

Part II: Entity Relationship Diagram

Case Study: Express Burger Restaurant



1. Based on the information provided in the previous scenario in italics, what entities will Express Burger need to store information about?

The ERD Diagram displays all the entity relations that are crucial for the successful operation of the restaurant. To start off, the 'Customer' entity would include essential details like names, emails, phone numbers, and billing addresses, allowing the restaurant to maintain a personalized relationship with each patron. The 'Orders' entity would record each transaction's specifications, including order ID, date, and total price. It is vital to track a customer's order history and establish a pattern in their purchasing behaviour. The 'Menu' entity would list all the available main dishes and side items along with their prices, ensuring that customers are informed about what they can order. The 'ItemOrder' entity links the orders to specific menu items detailing the quantity of each item in an order. This is where the unique requests for express mighty meals would be tailored, noting the combinations and the sizes meant to feed varying numbers of individuals. The 'Inventory' entity is key to keeping track of all the ingredients necessary for the preparation of the meals, while the 'Supplier' entity holds information on the sources of these ingredients, ensuring a steady supply and enabling effective inventory management.

2. For the entities identified in the previous part, identify a set of attributes for each entity as well as specify an identifier for each entity.

Entity:	**Customer**	**Orders**	**Menu**	**ItemOrder**	**Inventory**	**Supplier**	**Contracts**
Attributes:	Name, Email, Phone Number, Billing Address	Date, Total Price	Name, Price	Quantity	Name, Quantity	Supplier: Attributes: Name, Contact Info Identifier: Supplier ID	Contracts (Entity is used to store formal agreements): Attributes: Contract Terms, Contract Status, Billing Cycle Identifier: Contract ID
Identifier:	Customer ID	Order ID	Item ID	ItemOrder ID (composite key consisting of Order ID and Item ID)	Inventory _Item ID	Supplier ID	Contract ID

3. What rules did you apply when selecting the identifier?

Four rules were applied when selecting the identifier: uniqueness, simplicity and stability, minimalism, and non-descriptive. To start off, the identifier must uniquely identify each instance of the entity, and there can not be two instances that have the same identifier. Secondly, the identifier should be simple and should not change over the lifespan of the entity instance. Thirdly, the identifier should be as brief as possible, such as the use of an ID number. The identifier would not include real-world meaning. This means that the entity will maintain abstraction and prevent issues from potential changes to the real-world equivalent entity.

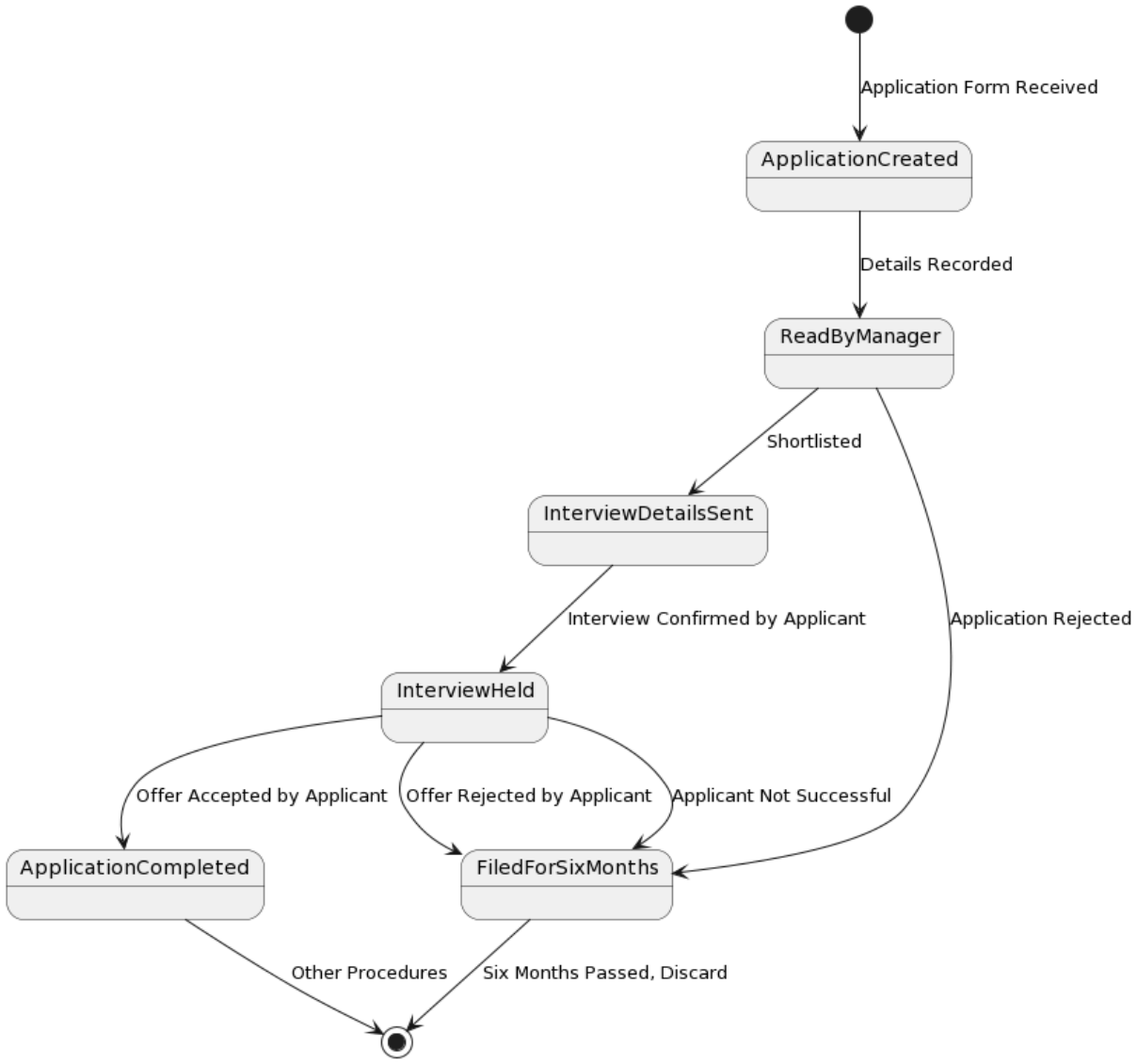
For example, 'Customer ID' is chosen as the identifier for the Customer entity because it is unique to each customer and does not change even if the customer's name, email, or address changes. It satisfies all the criteria for an effective identifier. The same logic applies to the identifiers for other entities like 'Order ID' and 'Supplier ID'. In the case of the 'ItemOrder' entity, a composite key is used as it represents a many-to-many relationship between Orders and Menu items, and each record is uniquely identified by the combination of the Order ID and the Item ID.

4. Draw the entity relation diagram and be sure to specify the cardinalities for each relationship.

Relation:	**Customer to Orders**	**Orders to ItemOrder**	**Menu to ItemOrder**	**ItemOrder to Inventory**	**Inventory to Supplier**	**Contracts to Customer**
Cardinalities:	One-to-Many (1:N)	One-to-Many (1:N)	Many-to-Many (M:N)	Many-to-One (M:1)	Many-to-Many (M:N)	One-to-Many (1:N)

Part III: State Machine Diagram

Case Study: Job Application



Using the created state diagram, the process begins when an application form is received. An application is created with the details from the form, and the details from the application form are recorded in the system. Then, the application is read by a manager, who will decide its fate, either to shortlist the application or to reject the application and file for six months. If the manager believes the application is promising, it is shortlisted. From here, interview details are sent to the applicant. The applicant receives the details of the interview and confirms they will attend the interview, which, henceforth, the interview takes place. There are three possible outcomes for the post-interviews. If the applicant is offered the position and accepts it, the application is completed, and "other procedures" begin. If the applicant is offered the position but declines, the application is filed for six months before being discarded. If the applicant is not successful in the interview, a rejection letter is sent, and the application is filed for six months before being discarded. Regardless of the reason for filing, whether it be the initial rejection, rejected offer, or unsuccessful applicant, after six months, the application is discarded. Overall, the state diagram visually displays the potential pathways that an application can take once it is submitted.