COS214 Practical Project

Task Two

Zethembe Danise, u20704209

Lerato Kgomoeswana, u22542354

Lwando Msindo, u21507067

Nokukhanya Ndlovu, u22768352

Khwezi Ntsaluba, u22515012

Future Phahlamohlaka, u22524798

Nthati Rampa, u20475102

System Requirements

The purpose of the given practical project is to produce a fully functional simulation of the daily goings-on of a restaurant. Based on the specifications, the Restaurant will consist of two main components: the floor and the kitchen. The floor consists of dining tables, an entrance to the Restaurant, and a waiting area. The staff responsible for the restaurant floor include (but are not limited to) the servers (who are assigned to serve specific tables), the maître d' hotel, and the restaurant manager (the latter of whom will handle complaints received from the customers).

The staff responsible for the kitchen are three chefs (each of which is responsible for preparing parts of the Customer's orders) and the head chef, who adds the finishing touches to the meal and ensures that the meal prepared is of high quality.

As such, the following requirements have been identified for the floor and kitchen section of the restaurant simulation:

Functional Requirements – floor

A maître d’ hotel is responsible for seating customers at the first available table. Since the Restaurant works solely on walk-ins, no preference is given for where the customers will be seated. Furthermore, the maître d' hotel may also approach customers at their tables to check in on their dining experience, gauge their mood, and handle any complaints that they might have regarding their dining experience – from the time taken to prepare the meal to the quality of the meal.

The tables should be able to seat any number of customers, up to a specified maximum. Tables can be combined to cater to large groups of customers. However, the combining of tables introduces spacing issues in the Restaurant.

The customers should be able to place orders with the waiter assigned to their table(s). Since this is a build-your-own meal restaurant, customers may order any part of a meal to suit their dietary preferences. The order has a state associated with it. This will be communicated between the floor and kitchen staff.

Should all the tables be (fully) occupied, a waiting area is to be provided for the overflow until such a time that a table becomes available. Customers seated in the waiting area may choose to leave if their expected waiting time exceeds the amount of time spent in the waiting area.

As expected from diners, the customers have temperaments influenced by their dining experience. Customers are allowed to complain to the waiter, which, depending on their severity, may be communicated to the maître d' hotel and, finally, the manager. The complaints will be resolved in a manner appropriate to the type of complaint and its severity. That is, customers may receive an apology, they may have their meal replaced with one of a higher quality, or their meal(s) will be complementary.

Customers may opt to start a tab with the Restaurant, or they may pay for their meals outright. The bill can be split up equally between the customers. That or different courses may be paid for by one (or more) of the patrons.

Functional Requirements – Kitchen

As specified earlier, there will be four chefs operating in the kitchen, three of which have a specialty for the preparation of the customers' meals:

One chef is assigned the task of handling fried delicacies. Another chef is assigned the task of preparing burgers (of which there are chicken and beef) burgers. Finally, we have a chef responsible for preparing poultry and bovine foods.

The head chef is responsible for adding the finishing touches to a meal. This may include (but is not limited to) garnishing the meal, preparation of delicacies that fall within the head chef's specialty, and ensuring the quality of the meal produced.

At any given point during the preparation of the meals, the chefs may receive complaints from the waiters, maître d' hotel, or the manager. The chefs will handle these complaints in a manner appropriate to the nature of the complaint. Meals with a lengthy preparation time warrant an apology and a request for patience from the customers who ordered the meal. Meals of poor quality will warrant the preparation of another meal, which will be at the expense of the chef.

The chefs will alert the waiters once the orders have been prepared if the chefs have completed the preparation of the orders they have been tasked with.

End-user Requirements

Given the functional requirements, several end-user requirements have been identified for both the floor and kitchen sections, as well as the staff operating the sections of the Restaurant. Note that the responses to events in the simulation will be shown in the terminal.

Floor End-user Requirements

The floor section of the Restaurant should be instantiated, during which the user will be prompted to choose an arbitrary number of tables where the customers can be seated.

The maître d' hotel should seat the customers at the appropriate table(s). That is, at a dining table if tables are available or in the waiting area if all the tables are (entirely) occupied. The user will see this functionality in action when the Customer.

The waiters are instantiated along with the tables, and they are assigned IDs that match the number of the table that they have been assigned to serve. The user will find the customers' readiness to order will be randomized such that the waiter will then be prompted to take the customers’ orders or will continue serving other tables until such a time that the customers are ready to order.

The waiters, manager, and maître d'hotel will visit each of the tables, enquiring as to the customers' disposition and gauging the quality of their dining experience. The user will see this visitation in action in the form of customer responses, which, depending on their mood, will be a complaint or compliment, triggering a response from the waiter, manager, or maître d' hotel.

Kitchen End-user Requirements

It should be noted that the requirements of the kitchen are intertwined with those of the floor section of the Restaurant since the chefs receive feedback from either the waiters, manager, or maître d' hotel and will respond accordingly.

The details of the kitchen, from the preparation of the meal, will be made available to the user. The user will be able to follow the stages of preparation of the meal. The meal will pass through the different chefs depending on which parts have been ordered. Finally, the waiters will be alerted once the meals have been prepared. The customers will be served.

Activity Diagrams

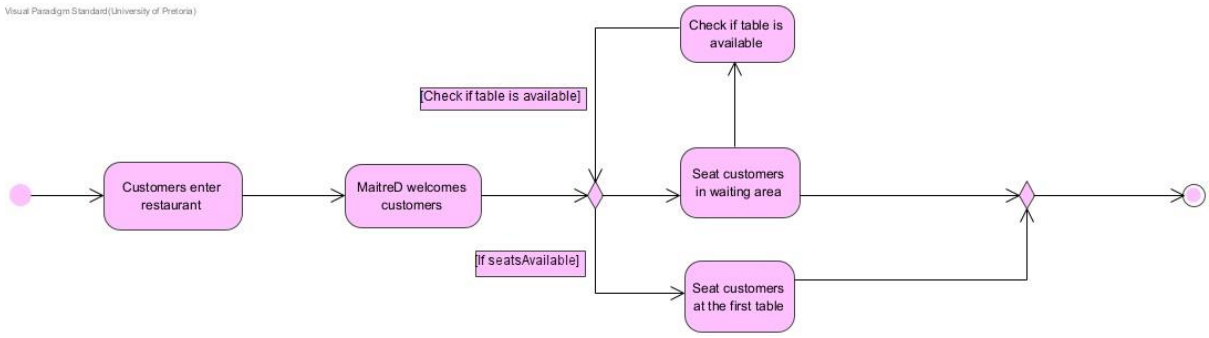


Figure 1: The maître d' hotel welcomes and seats the customers.(Above)

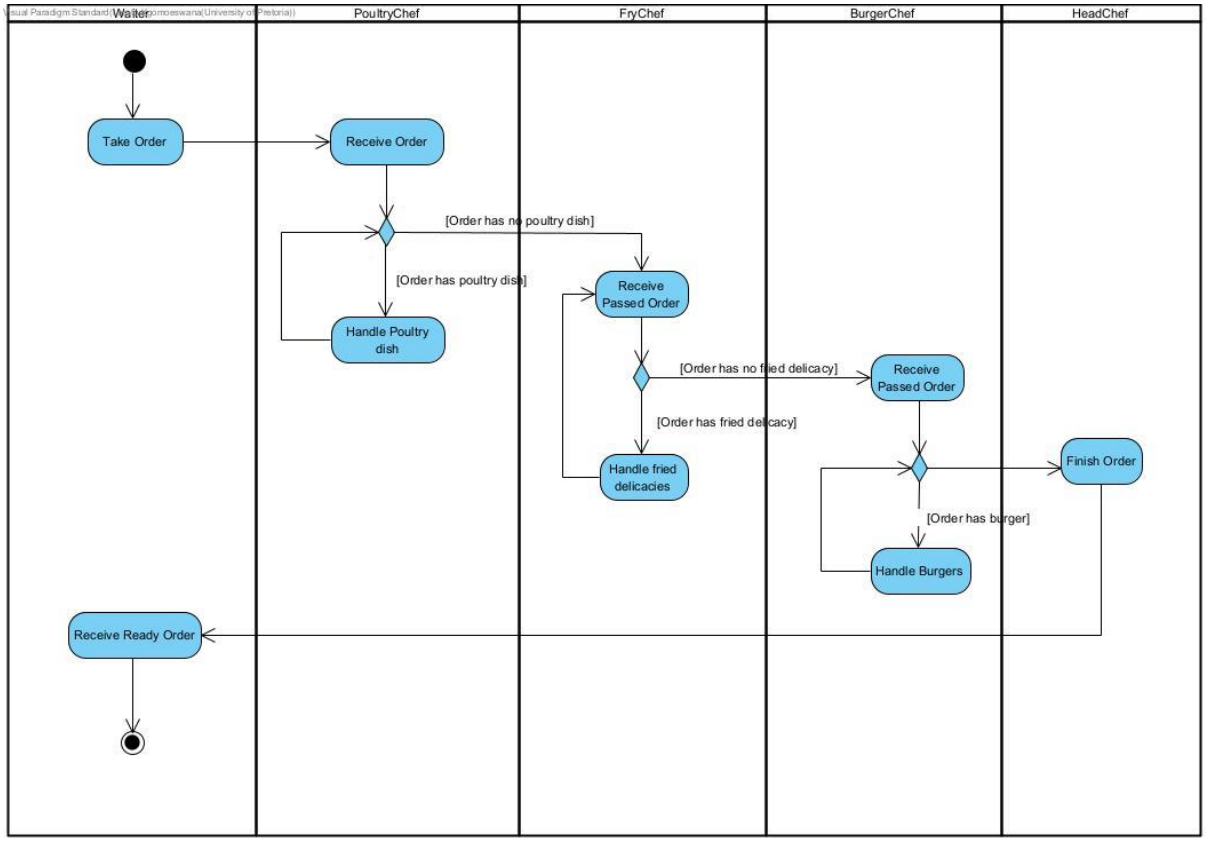


Figure 2: The diagram models the waiter-customer interaction as well as the preparation of the meal. Once the meal is prepared, it is taken to the patrons. (Above)

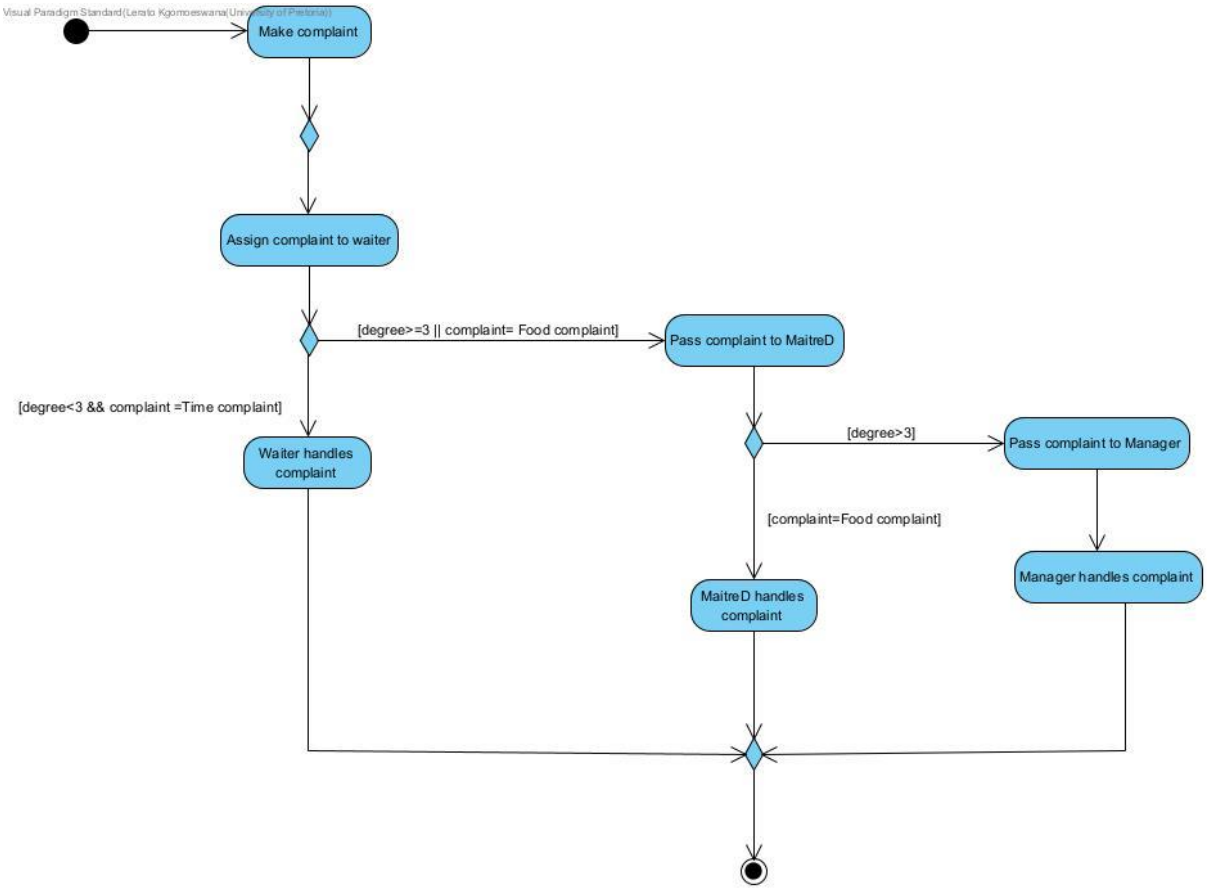


Figure 3: The complaints system. Depending on the type of complaint and its severity, the complaint will be handled by the appropriate restaurant staff. (Above)

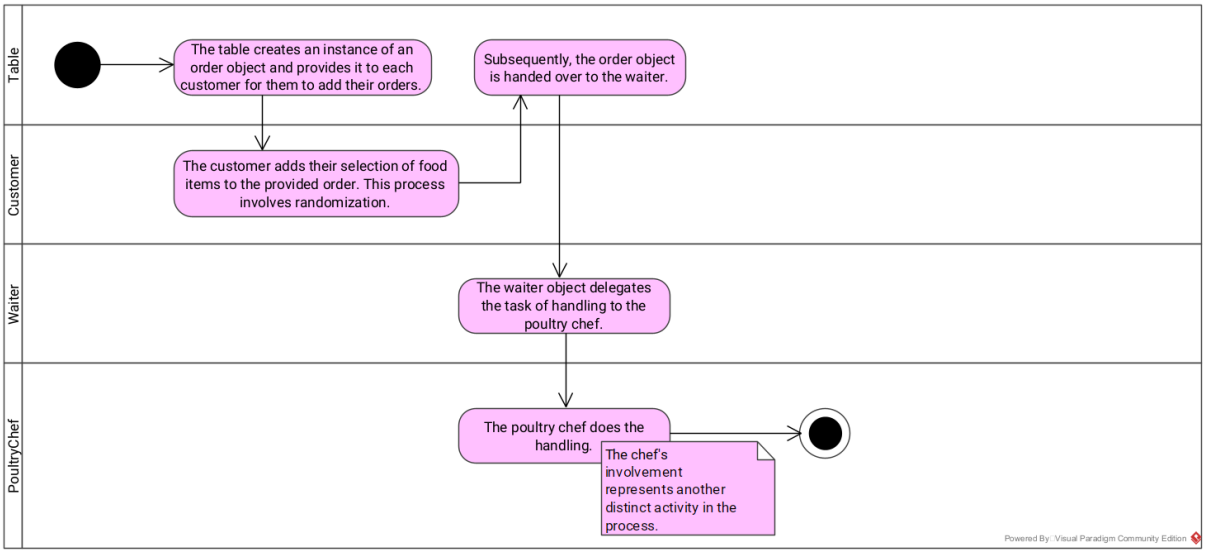


Figure 4: A description of a particular order to be prepared by a specialty chef. (Above)

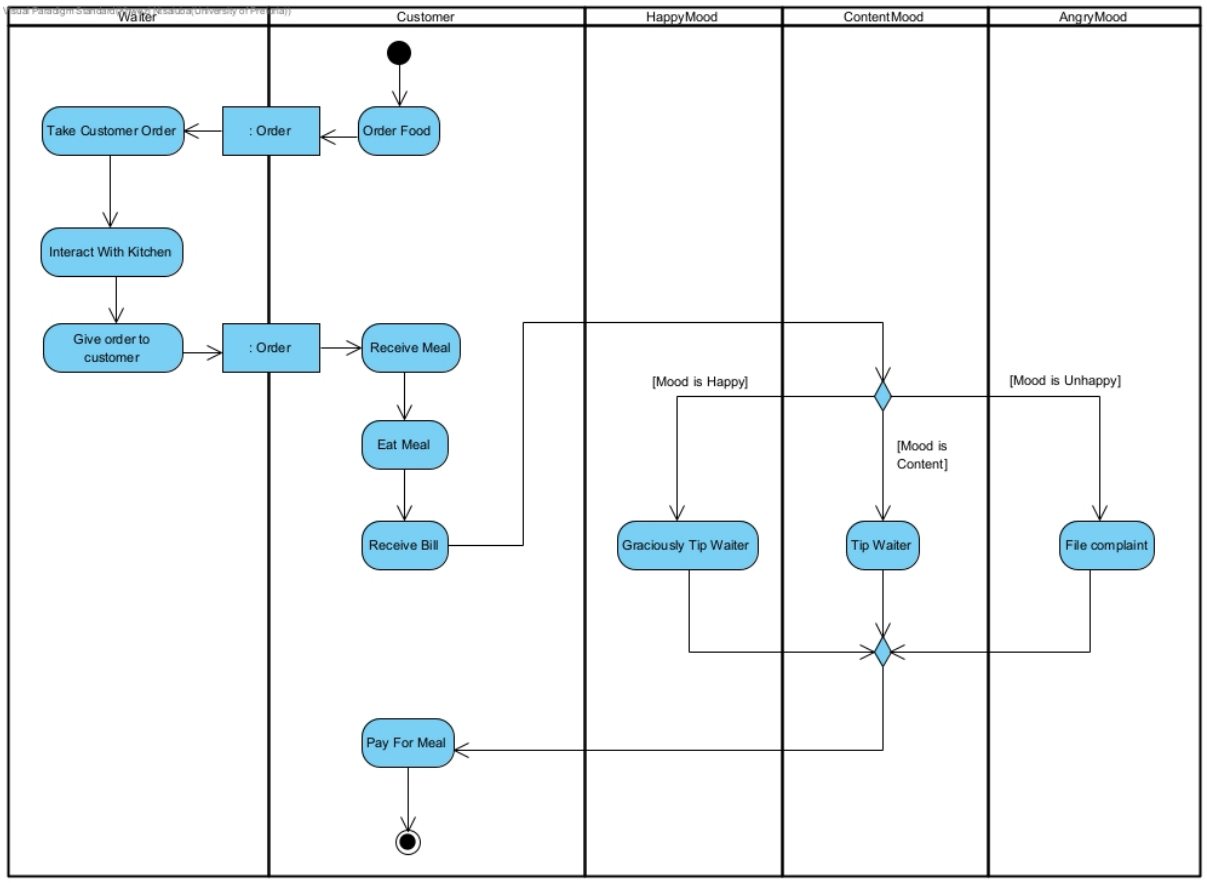
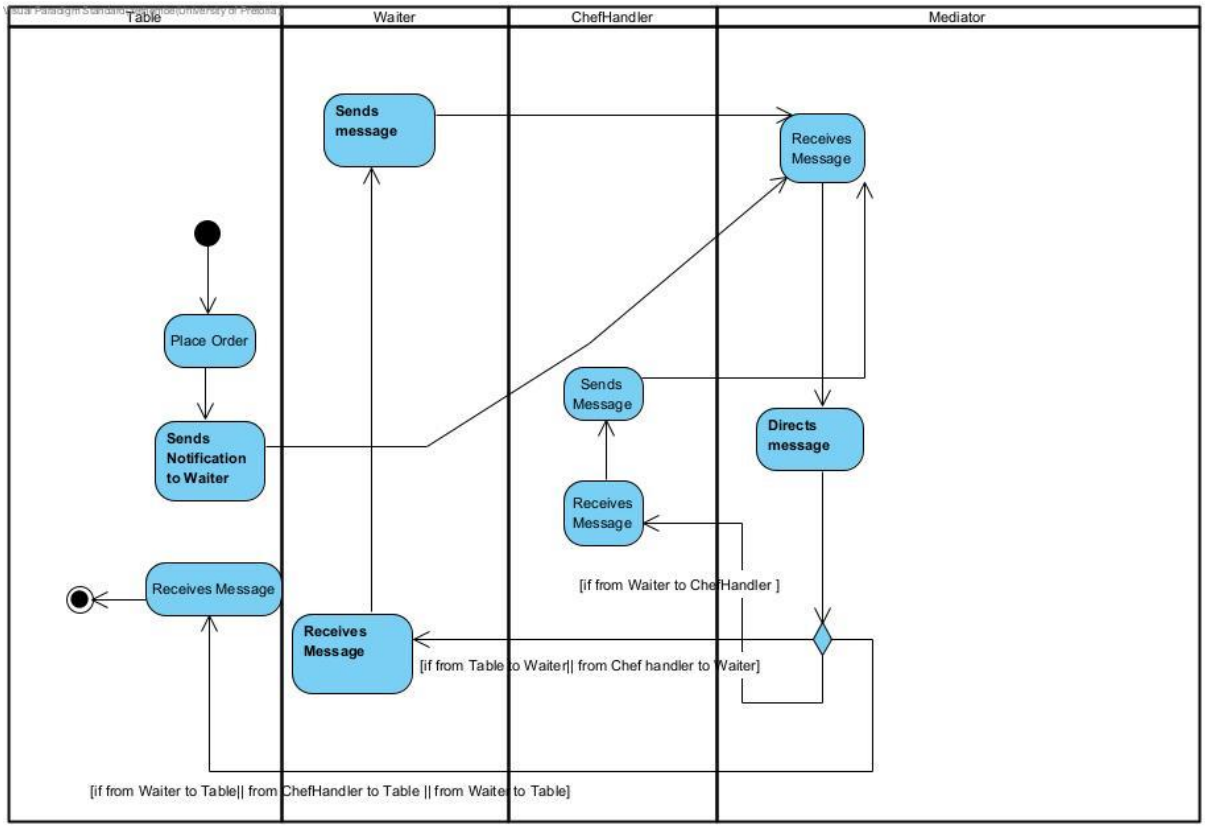


Figure 5: The payment system. Customers may tip depending on their mood, which is influenced by their mood. (Above)

Figure 6: A description of how the messages are sent to different subsystems in each System, the floor, and the Restaurant. (Below)



Based on the functional requirements described, as well as the processes modeled by the activity diagrams above, the following design patterns have been identified as suitable for representing the components of the restaurant floor:

Singleton: A restaurant typically has one maître d' hotel, whose responsibility is to welcome and seat customers at a table. Seating is dependent on customer preference and the number of customers that need to be catered for.

Chain of Responsibility(Based on the Complaints): When customers place their orders, they are communicated to the waiter (or waitress) and then to one of the chefs in the kitchen. The complaints are conveyed to the maître d'. If the matter is still unresolved, it can be escalated and reported to the manager.

Chain of Responsibility(Based on the Chefs): The aim of the chain of responsibility is based on the chef subsystem; it coordinates the activities of multiple chefs in the kitchen, each with distinct roles in preparing various food items. Depending on the food items included in a customer's order, different specialized chefs, including poultry chefs, fry chefs, and the head chef, are assigned to handle and prepare these specific food items. The System ensures the efficient and timely preparation of meals while maintaining the highest quality standards.

State (Based on the Customer): The Customer's mood can be modeled using the state design pattern. The patrons' moods may shift depending on the dining experience, and their actions reflect the shift in their mood. As specified in the *Chain of Responsibility*, the customers can complain if they are unhappy or dissatisfied, or they may tip if the service is excellent. Furthermore, the customers' readiness to place an order can be modeled utilizing states.

State(Based on the Order): Based on the order subsystem, the state pattern aims to solely keep track of the state of the order the chefs are preparing for the Customer. The order is initially set to the state raw. Then, when passed to the first chef- who can handle the order, the state of the order will then be changed accordingly to the InProgress state. Finally, when the order reaches the Head chef with the addition that the Head chef has approved the order being worthy to be severed, then the state of the order will change to Done.

State(Based on the Table): The table(s) will have the state associated with them that will change as time passes in the System. A table can have one of the following six states at any point in time in the System: EmptyTable, OccupiedTable, RequestToOrder, ReceivedOrder, AwaitingBill, and PaidBill. The goal is to employ a different behavior that will act upon the table object based on the current state of the table state. Each table state subclass specifies its successor state.

Iterator: If a waiter is responsible for taking the orders of multiple tables, an iterator would be appropriate. The waiter moves from table to table (depending on the

Customer's readiness to order), takes the customers' orders, and passes them on to the kitchen staff.

Mediator: The mediator is responsible for keeping track of subscribers and colleagues who have registered and want to communicate through the mediator to their colleagues. This is done by calling the register function, which will, in turn, add the colleagues to the "Database" or the respective vectors -solely for the mediator to gain access to the respective colleagues. The mediator is also responsible for ensuring that the right message reaches the correct recipient (specified by the sender), ultimately triggering the receiver (Object) to change its behaviour or perform some action. The mediator pattern in this subsystem mediates the communication between the following objects: The waiter, the Table, and lastly, the Chef Handler. This pattern aims to promote loose coupling between objects that would regularly interact.

Composite: Employed for adding different attributes/extra features to the food items, for example, adding chip sauces to the plain chips. This pattern allows for dynamically adding new behaviors or responsibilities to individual objects without affecting the behavior of other objects of the same class.

Decorator: Utilised for different burgers (e.g., chicken burger and beef burger) as each type has different patties. This pattern defines the skeleton of an algorithm in the superclass but will allow subclasses to override specific algorithm steps without changing its structure.

Template Method Design Pattern: Utilised for different burgers (e.g., chicken burger and beef burger) because each type has different patties. This pattern defines the skeleton of an algorithm in the superclass but lets subclasses override specific steps of the algorithm without changing its structure.

Strategy Pattern: Applied for different cooking methods for various food items (poultry, burgers, chips). This pattern enables the selection of an algorithm at runtime, such as choosing between grilling or frying, without tightly coupling the client code to the specifics of the algorithms.

Façade: With the various subsystems that, in turn, employ their system requirements, the façade was chosen to introduce a high-level interface that will mask the complexities of the System as a whole. This will allow the end user to have a smooth, simple interaction with the System with a simple push of a few buttons.

Object Diagrams

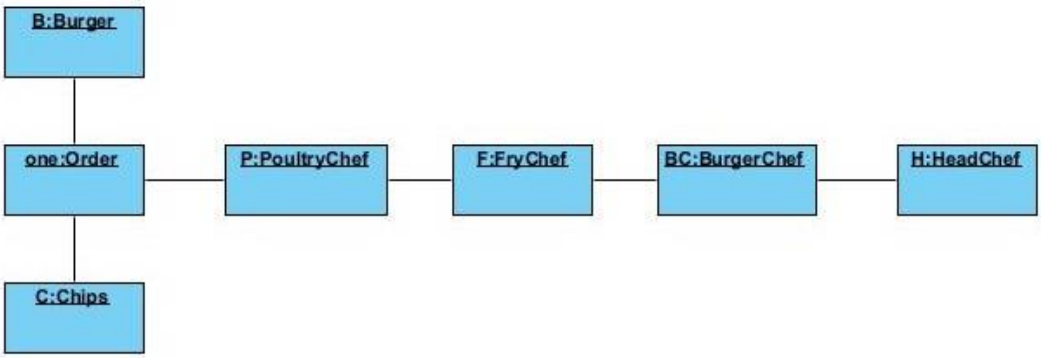


Figure 7: The specialty chefs The interactions between the kitchen staff when handling orders (Chain Responsibility) (Above)

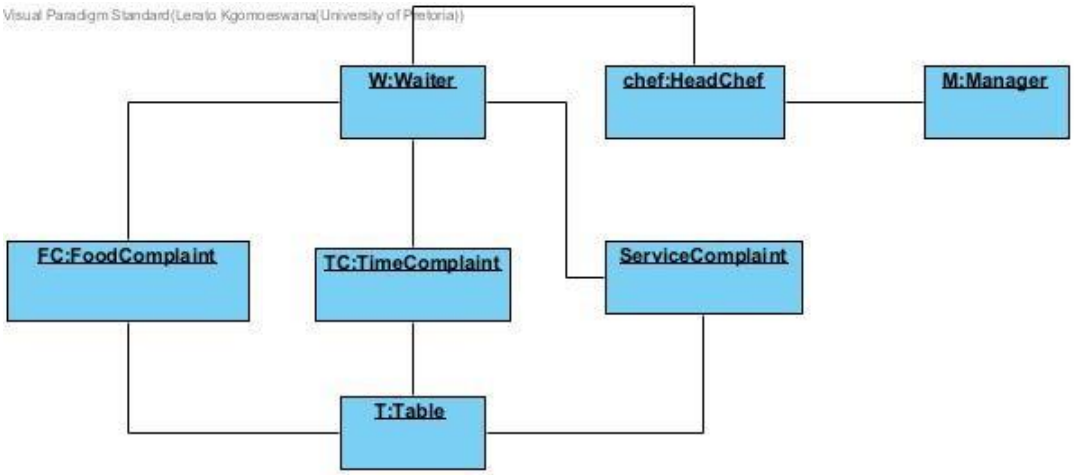
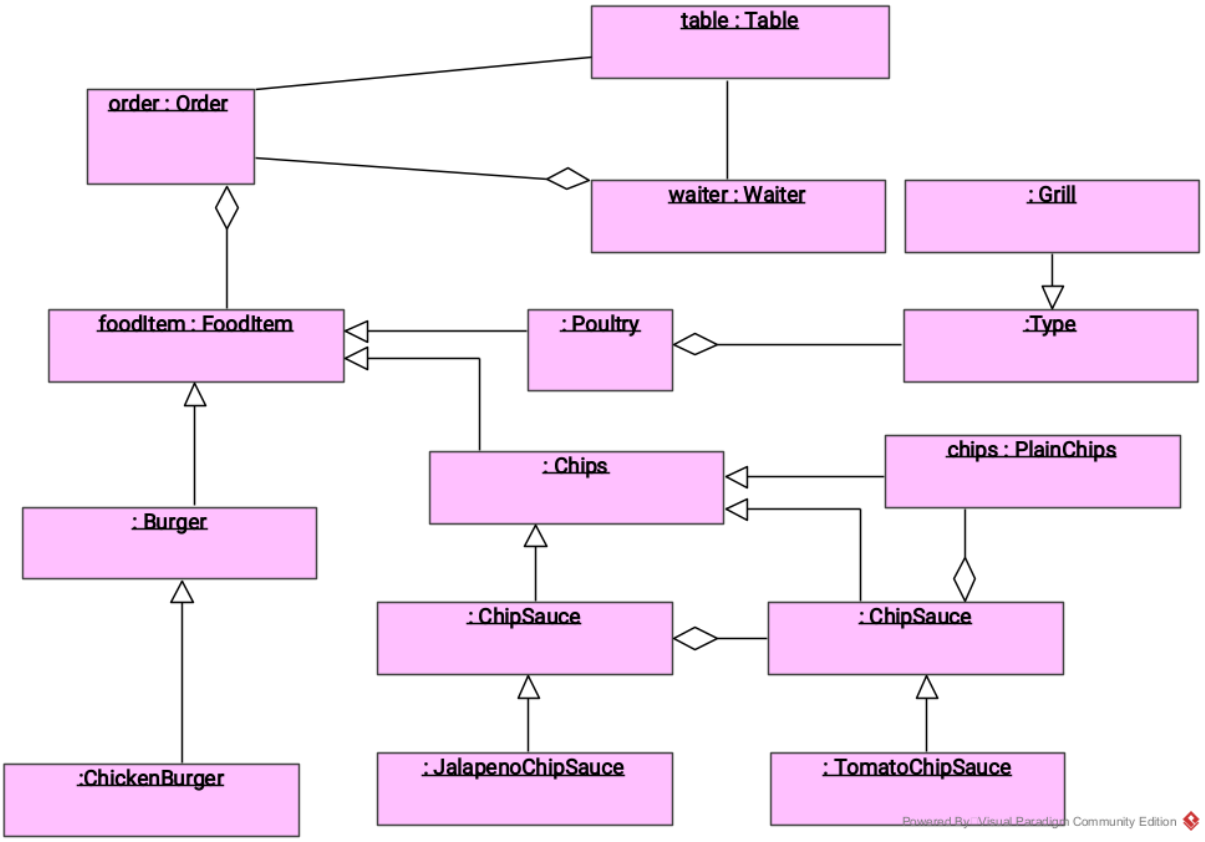


Figure 8: The interactions between the kitchen staff when handling complaints(Chain of Responsibility). (Above)

Figure 9: Object diagram showing the relationships between the food classes associated with the kitchen section. (Below)



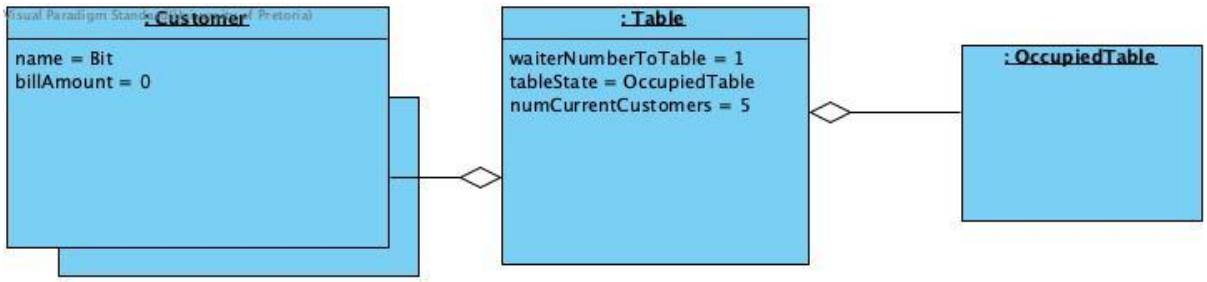


FIGURE 10: Object diagram showing the relationship between the Customers, Table, and the state of the table. (State pattern) (Above)

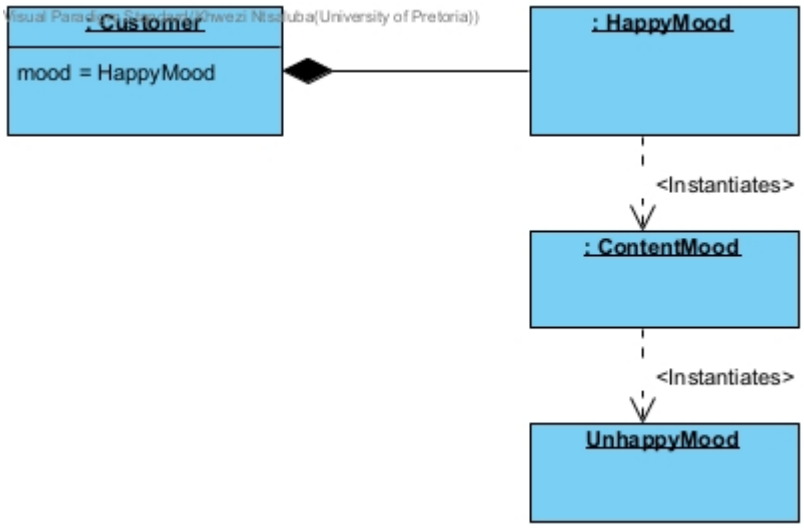


Figure 11: Object diagram showing the relationship between the Customer and the Concrete moods it interacts with (ConcreteStates) -State pattern. (Above)

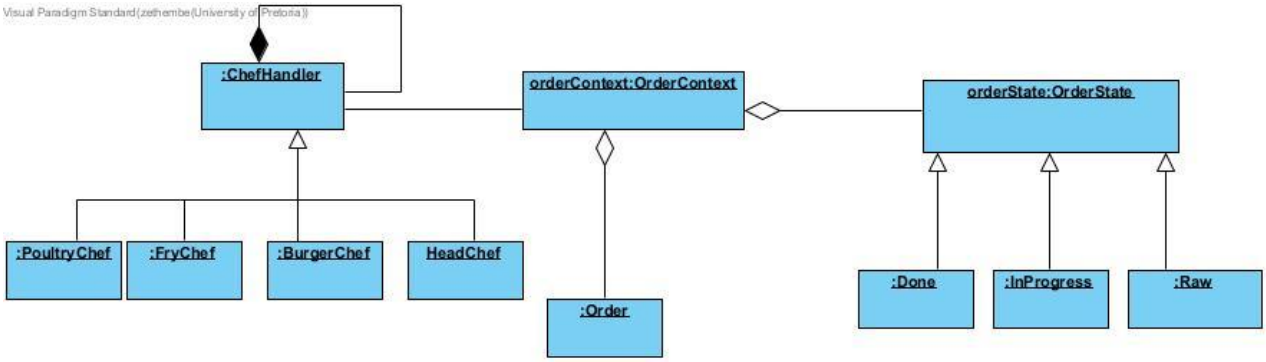


Figure 12: Object diagram showing the relationship between the Concrete Order (ConcreteStates) classes, Order Context (State Context), and the ChefHandlers (ConcreteHandlers)-State pattern. (Above)

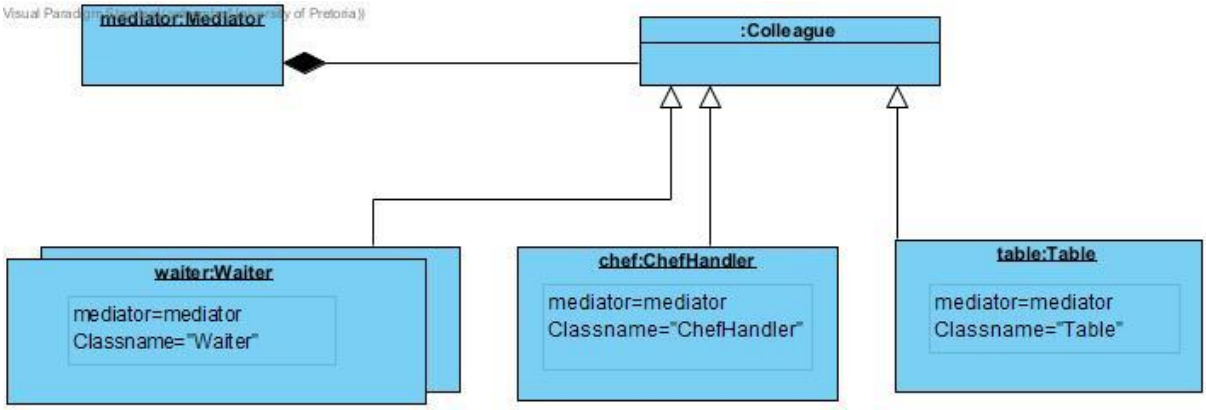
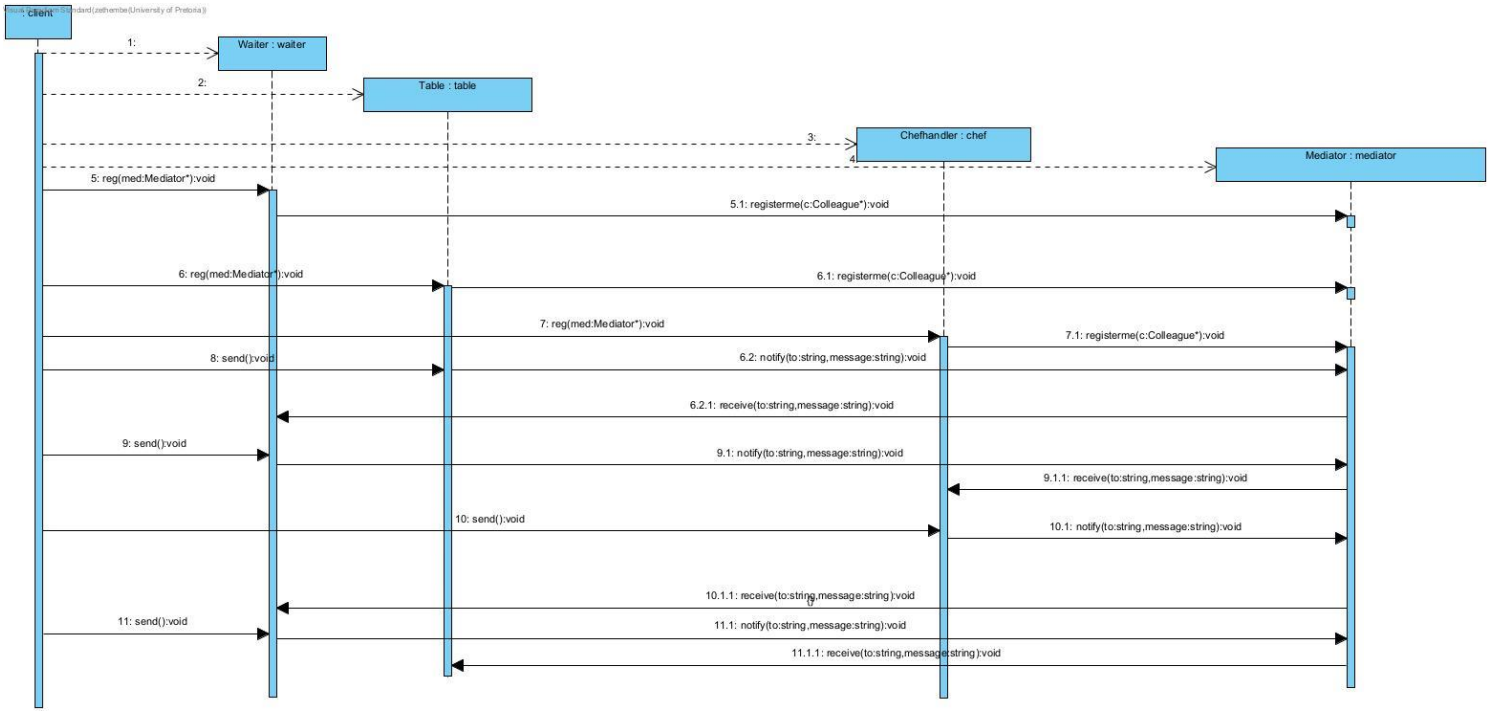


Figure 13: Object diagram showing the relationship between the Waiter, Table, ChefHandler(Concrete Colleagues), and the Mediator-Mediator pattern. (Above)

Sequence Diagram for message passing in System



State Diagram showing the change of state of an object in the System over

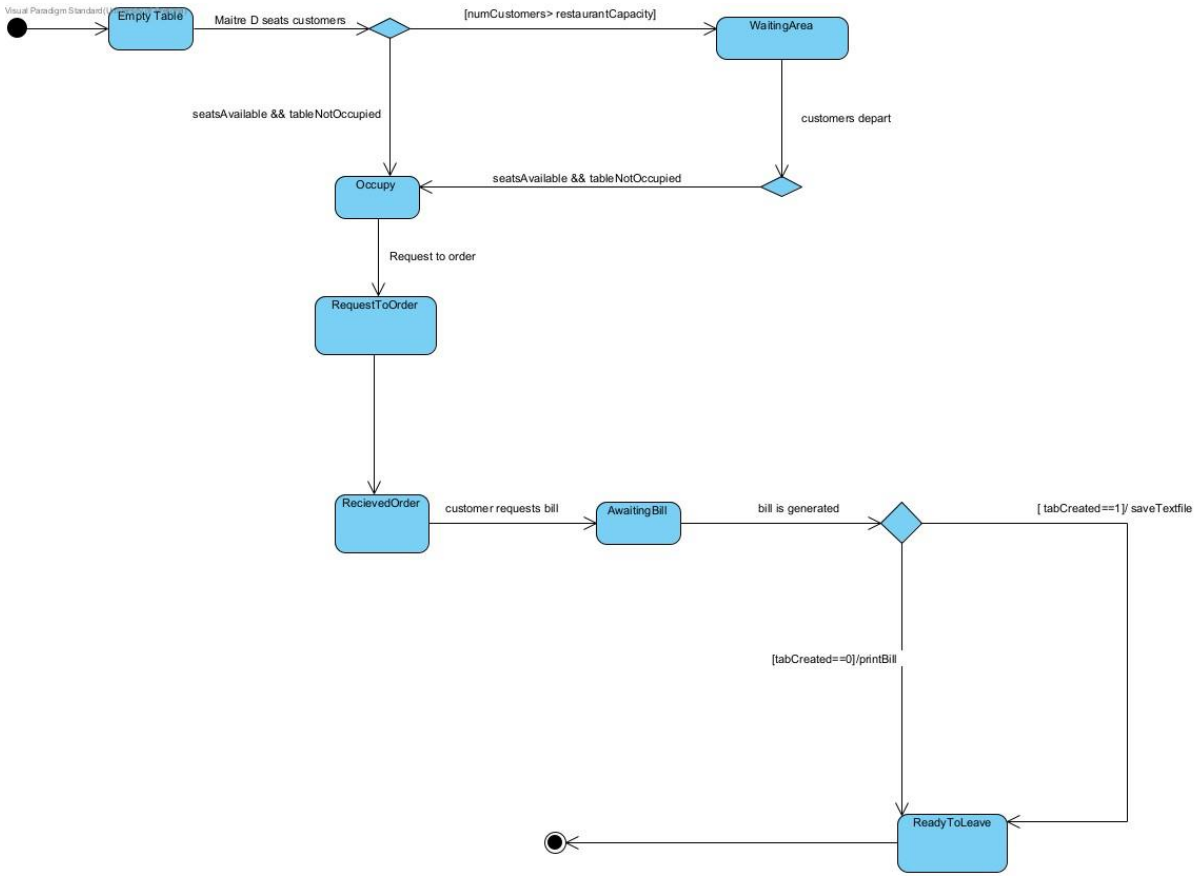


Figure 6: A succinct description of the primary operations performed in the simulation. (Above)

Communication Diagram for message passing in System