Take A Byte

Group Members:

u18234039 – Jiahoa Hu

u21439321- Sonaly Ambelal

u21508322- Zandile Modise

u21516741- James Fitzsimons

u21525936- Victor Zhou

u21556416- Asakundwi Siphuma

u21785742- Itumeleng Moshokoa

GitHub Repository: https://github.com/ZandileModise/PA5-214-Repo

Task 2: Design

System Requirements:

- Our system should simulate the operation of a restaurant which includes managing tables, taking orders, and serving food.
- Customers must be able to make orders and specify preparation methods.
- Waiters should interact with customers, submit orders to the kitchen and serve prepared dishes.
- The kitchen staff i.e., the chefs should prepare food orders and should work at different stations.
- The system should handle customer expectations.
- Bill management should include splitting the bill and starting tabs.
- The restaurant may have a booking system or walk-in service.
- The restaurant may have a bar.

Functional Requirements

Need two main interfaces:

Floor and Kitchen

The floor consists of:

Waiter and Table Management:

- Waiters should be assigned specific tables for which they are responsible for.
- Each table will be assigned a waiter.

Queue Management:

- A queue management system must be in place for customers waiting to enter the restaurant.
- Customers in the queue should be organized and seated based on availability.

Customer Seating and Management:

- There are 10 tables that seat four patrons each.
- The system must allow customers to request a table for seating.
- The table seat reserve system will allocate table for customers automatically.
- The system must be able to show customers to their assigned tables.
- Tables can be combined or split as needed to accommodate different party sizes.

Ordering and Serving:

- Customers should be able to place orders with a waiter.
- Customers may request the waiter to come back later if they are not ready to order.
- Waiters must pass customer orders to the kitchen staff.
- The kitchen must process orders as they come and delegate them to different chefs responsible for specific preparation tasks.
- Different stations in the kitchen should handle different parts of the dish preparation.

- Orders may be passed between stations and should be plated by the head chef before being served to customers.
- When an order is completed, the kitchen must notify the waiter for order pickup.
- The waiter will deliver the order to the customer's table.

Customer Expectations Management:

- The system should include mechanisms for managing customer expectations and feedback.
- It should allow for the tracking of customer satisfaction and the handling of customer complaints.

Build-Your-Own Ordering Menu:

- The system must provide a table of patrons a menu with a wide selection of available options, including appetisers, main courses, side dishes, and desserts.
- Customers should have the ability to create their orders by selecting items from the menu and customising them.
- The system displays real-time pricing information as customers build their orders, including the cost of individual items and any additional charges for customizations.

Waiter Actions Based on Table State:

- Waiters must be able to view the state of each table they are responsible for.
- The system should guide waiters in taking appropriate actions based on the table's state. For example, when a table is in the "Order Taken" state, the system may prompt the waiter to send the order to the kitchen.
- Waiters should have the capability to mark specific customer requests or table conditions, such as "Customer Not Ready to Order," for later review.

The kitchen consists of:

Order Processing:

- The kitchen system must receive orders from the waitstaff, specifying the items and special requests.

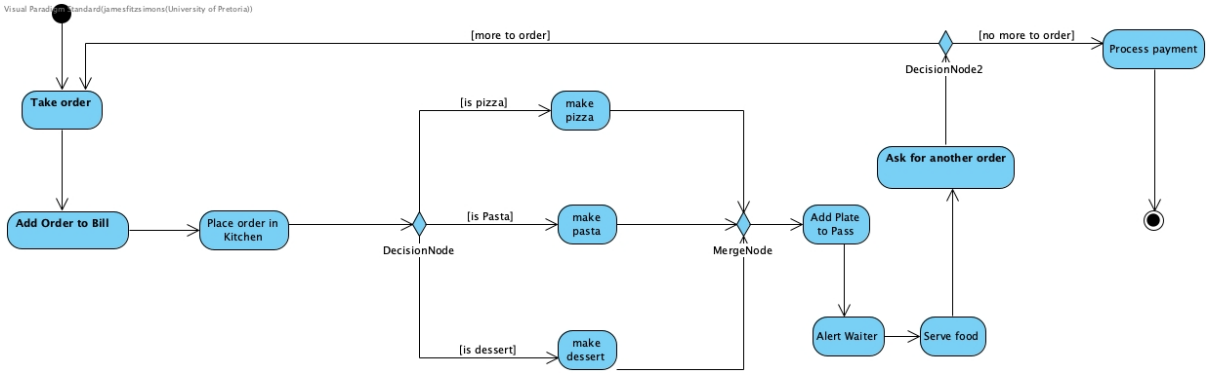
- Orders must be processed in the order they are received, following the first-come, first-serve principle.
- The system must delegate orders to the appropriate chefs responsible for different parts of the preparation process.

Chefs' Responsibilities:

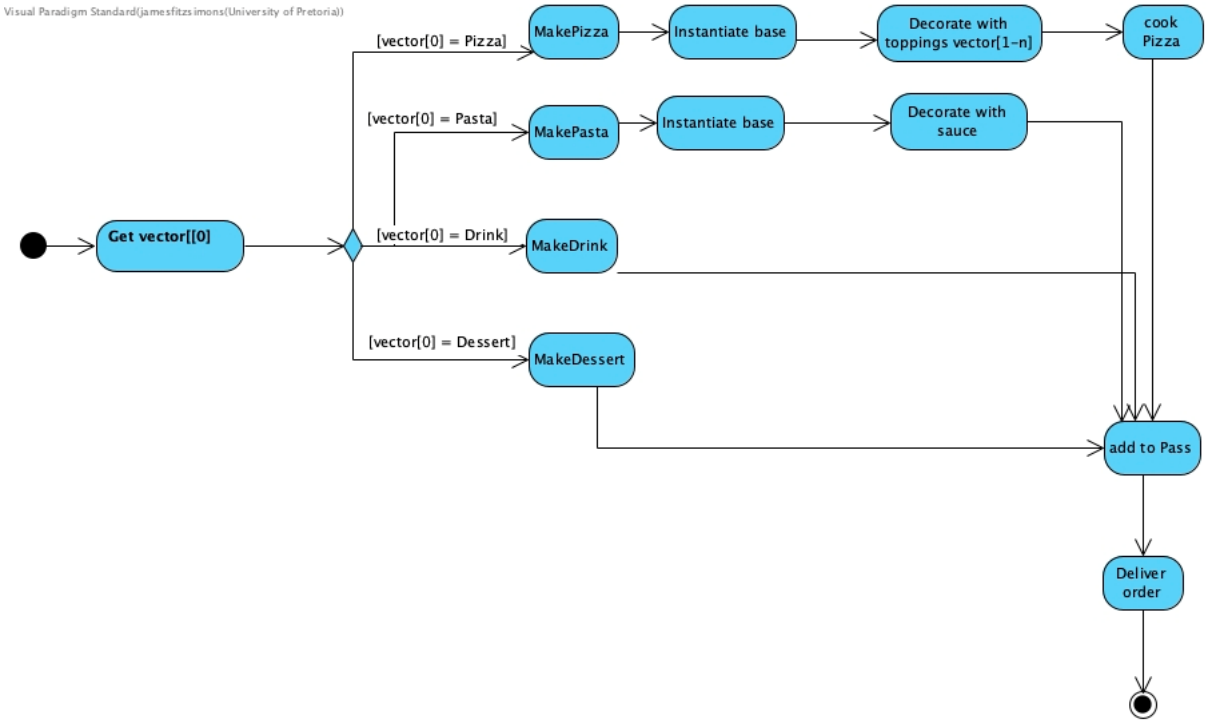
- Different chefs should be assigned specific tasks within the order preparation process (e.g., head chef for final plating, pasta chef for preparing pasta).
- Each chef should have access to the orders assigned to them and related details.

Activity diagrams:

Process following order placement:



Building an order:



Design Patterns

Floor

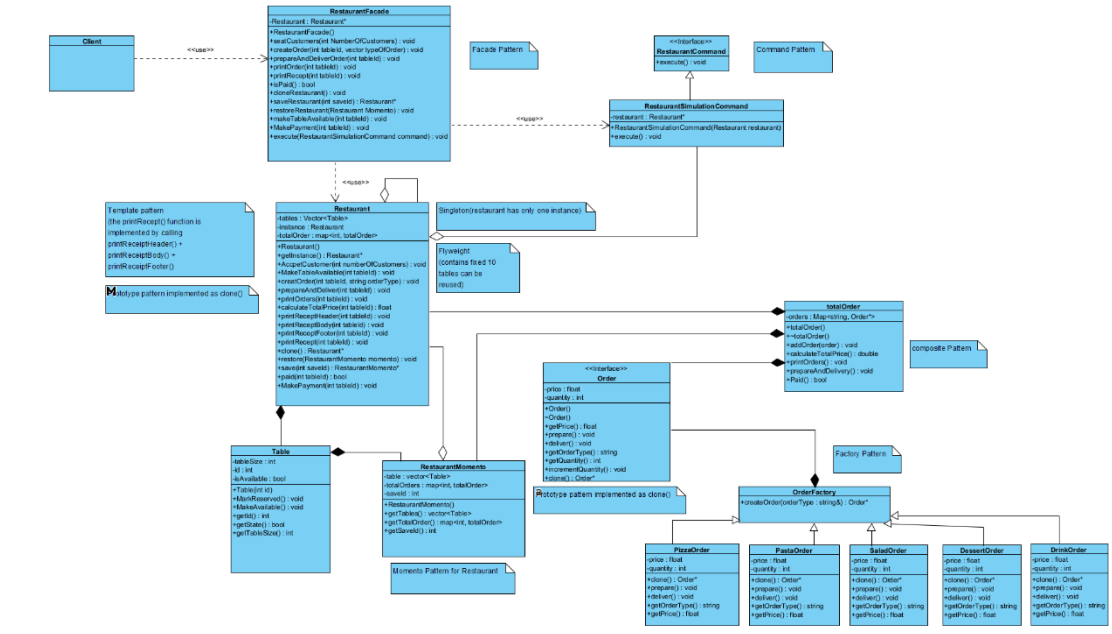
- Prototype
    - Clones the waiter objects.
- Memento
    - Saves and restores orders from clients.
- Singleton
    - The restaurant class serves as a singleton.
- Facade
- Flyweight
    - The kitchen class serves as a flyweight.
- State
    - Customer (waiting, order, served, eating, fed, complain ,exit)
    - Bill (paid, unpaid)
- Composite
- Iterator
    - Handles the customer queue.
- Factory
- Template

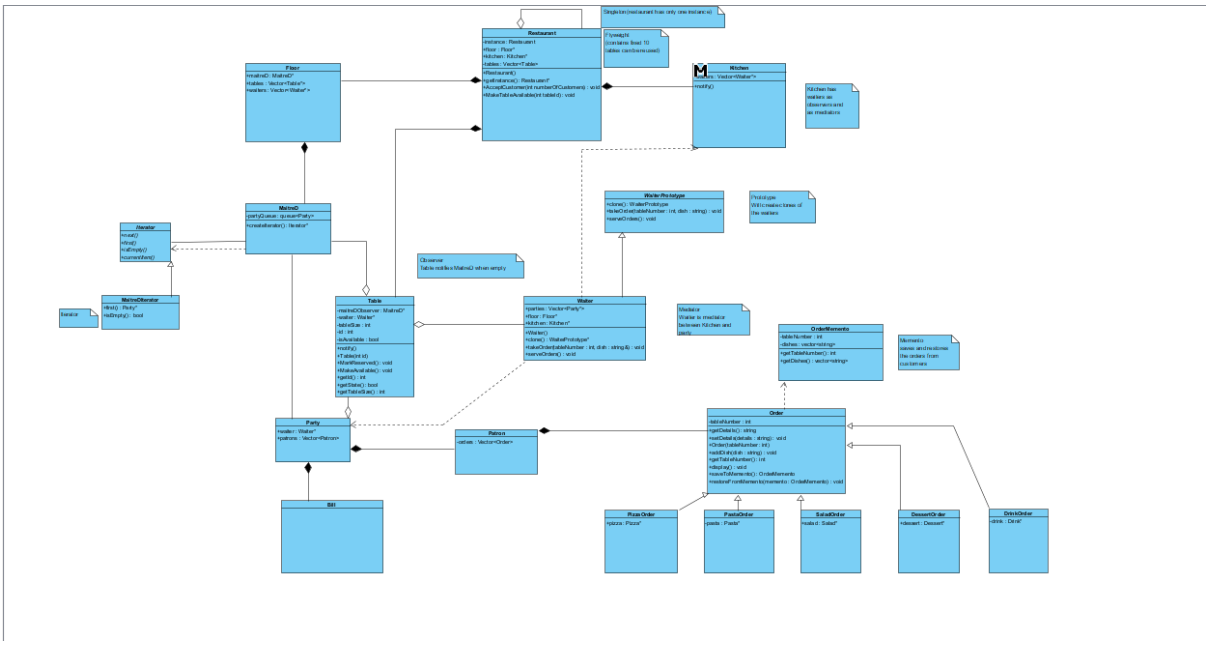
Kitchen

- Template
    -
- Decorator
    -
- Chain of Responsibility
    - Delegating the food to different chefs to prepare the dishes. (PizzaDoughChef ->PizzaSauceChef->PizzaToppingChef->CheeseChef)

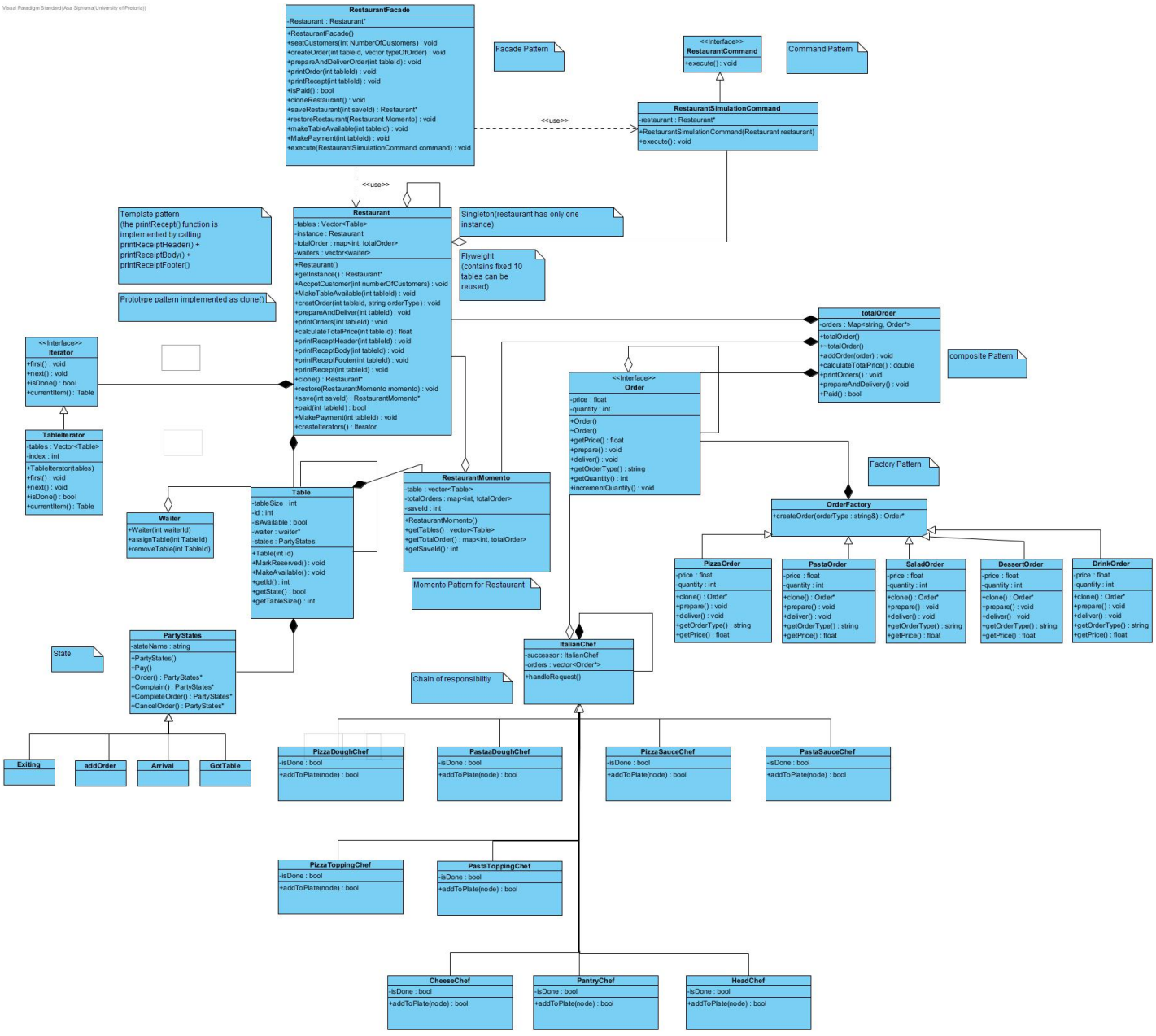
Complete UML Diagram

Initial floor UML

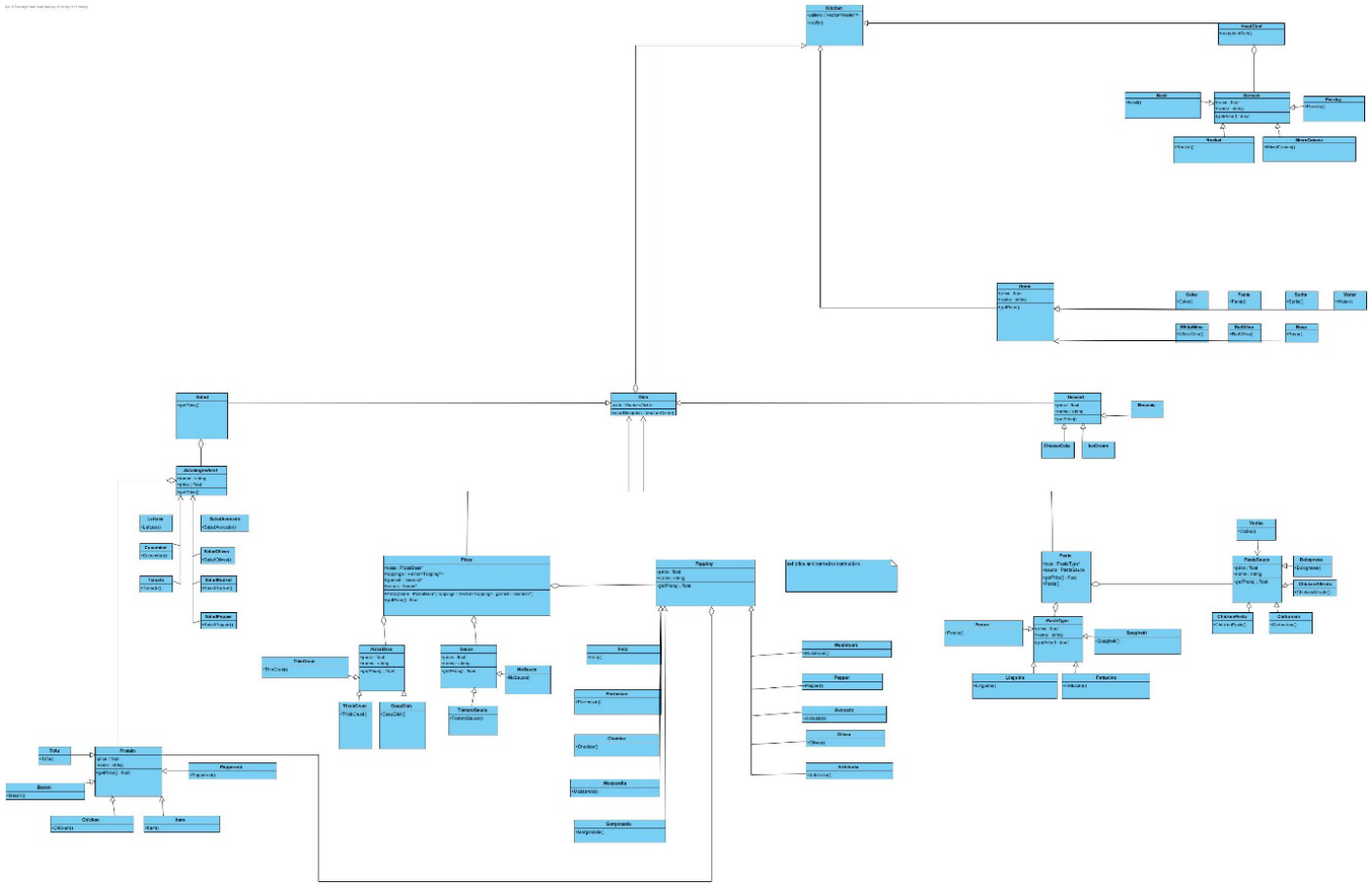




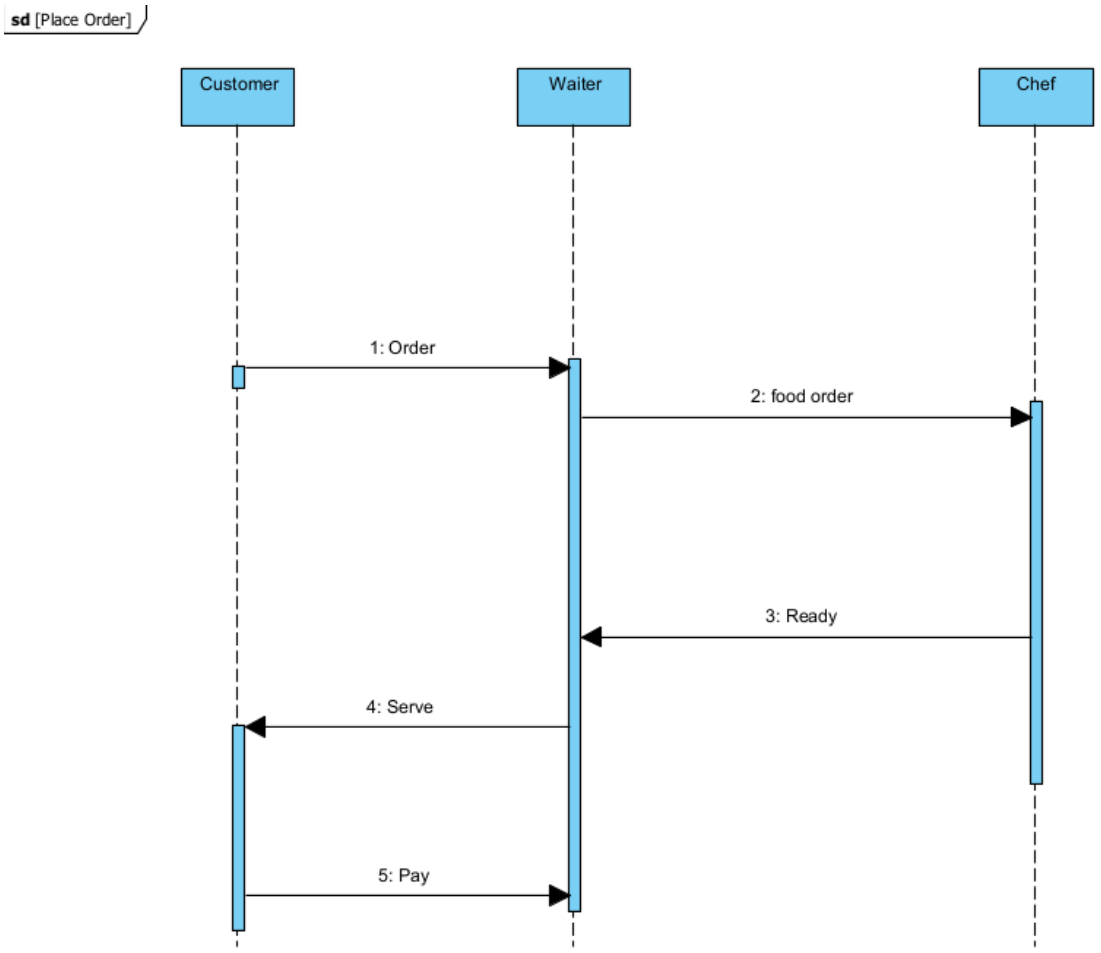
Final Floor UML



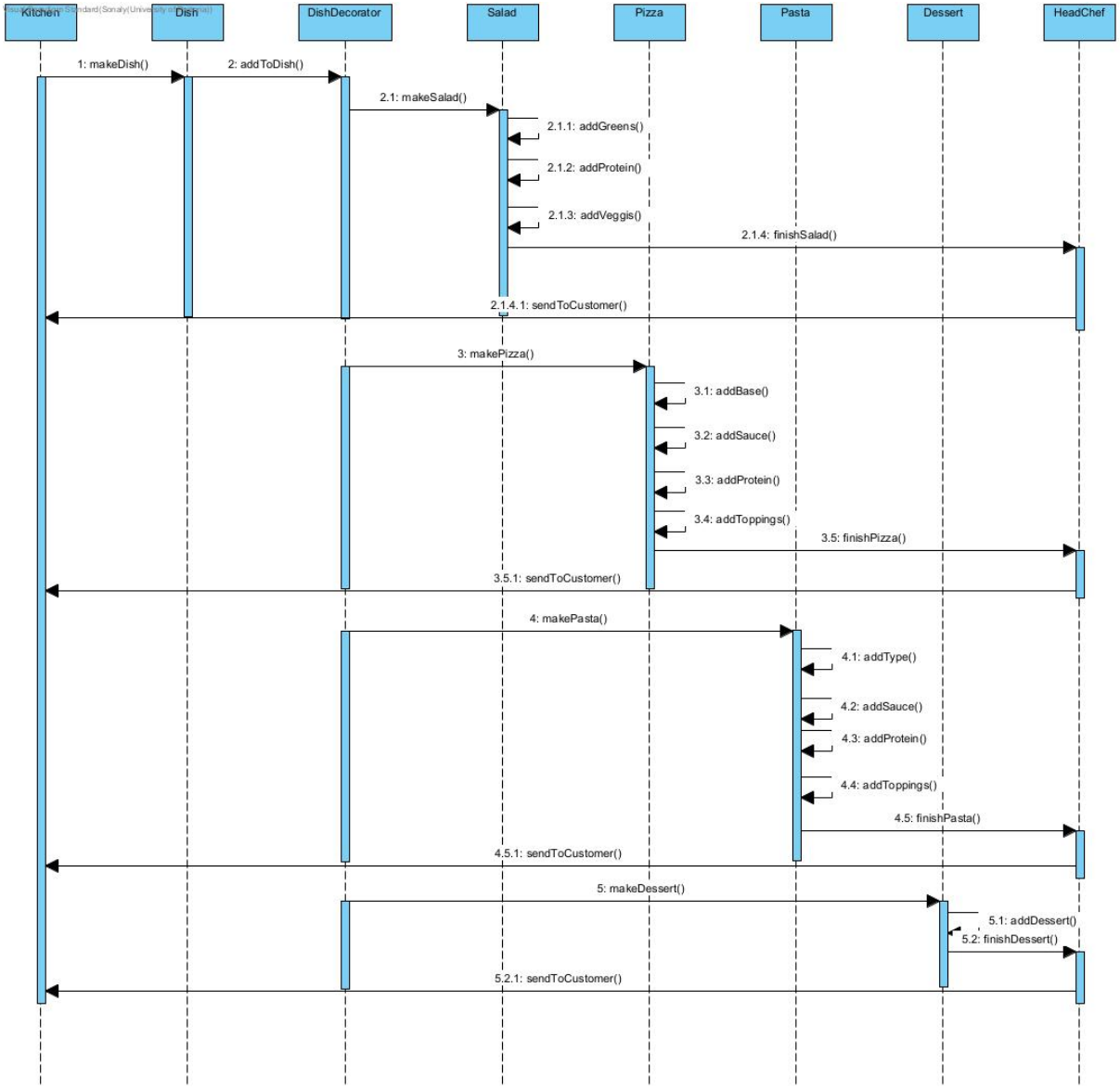
Kitchen UML



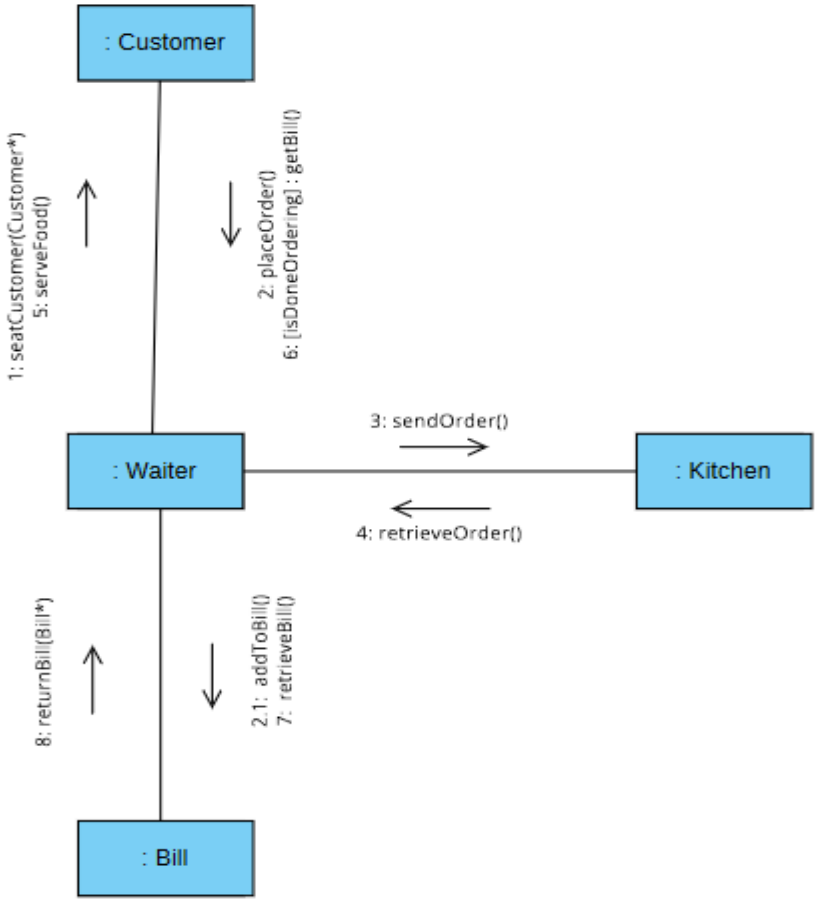
Sequence Diagram: (Place Order)



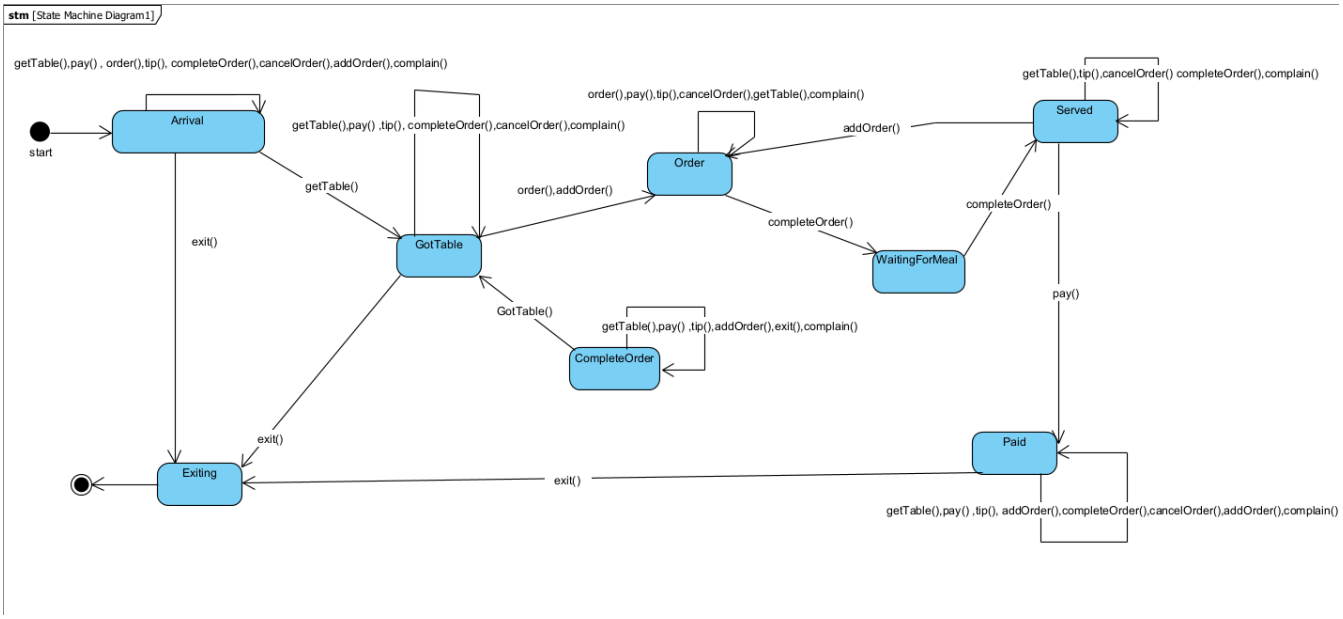
Sequence Diagram: (Kitchen)



Communication Diagrams:

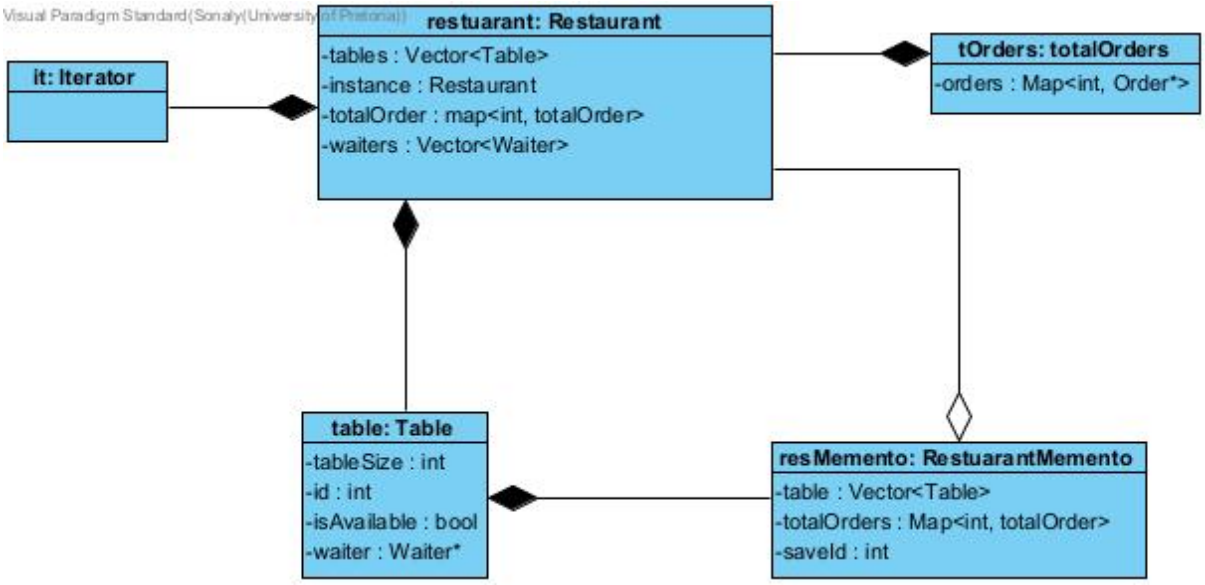


State Diagram

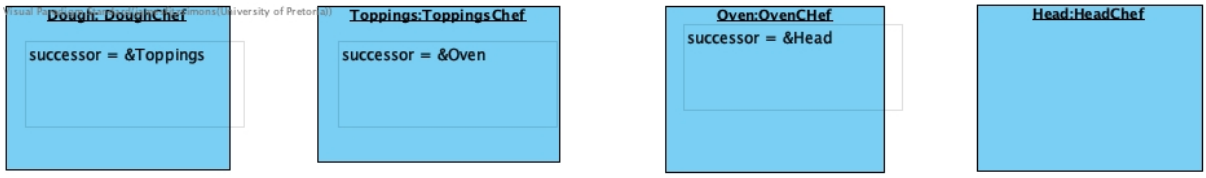


Object Diagrams:

Restaurant Object Diagram:



Chefs Object Diagram:



Draft Chain of Responsibility: