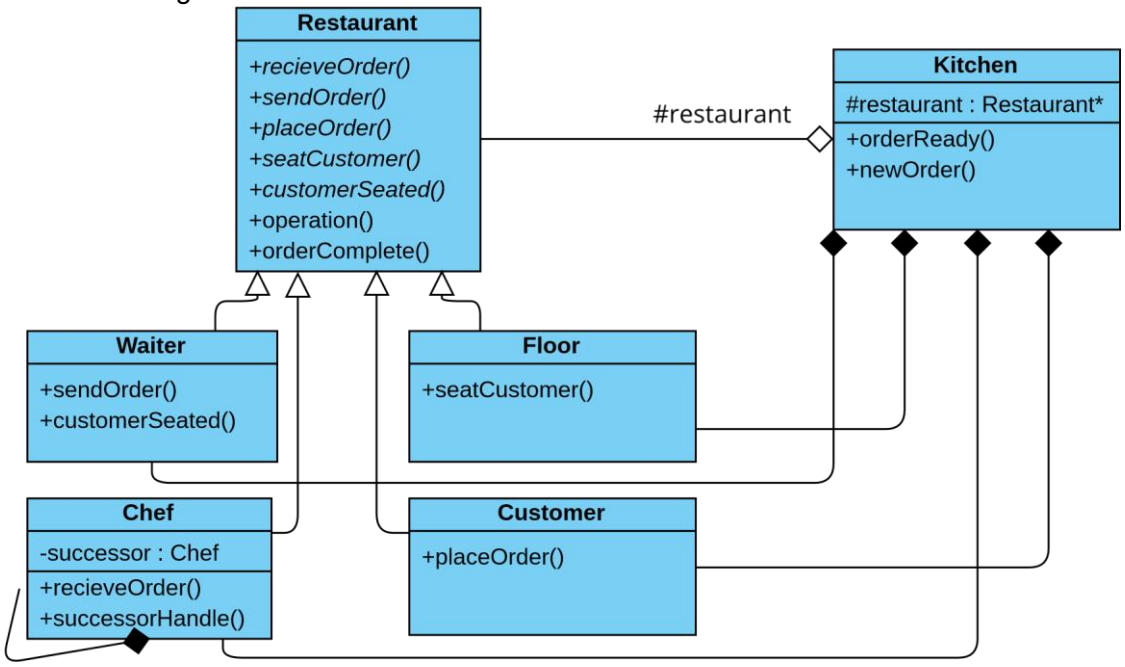
COS 214 Practical 5 PDF

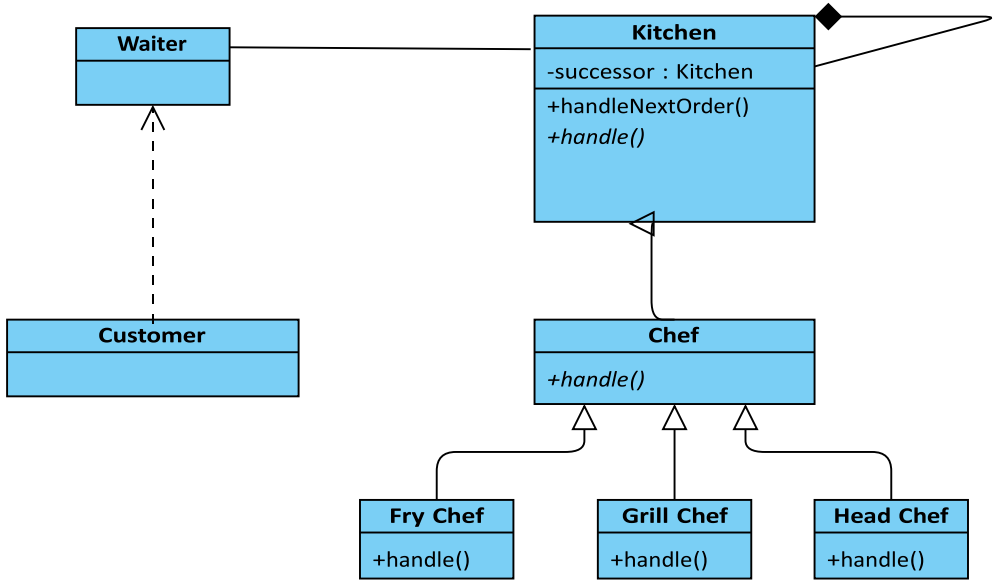
Group members:

- u21797545 - Mr. Douglas Porter
- u21479748 - Miss. Sange Tshakumane
- u14046492 - Mr. Mahleka Chuene
- u22625462 - Mr. Graeme Blain
- u22738917 - Mr. Aidan Chapman

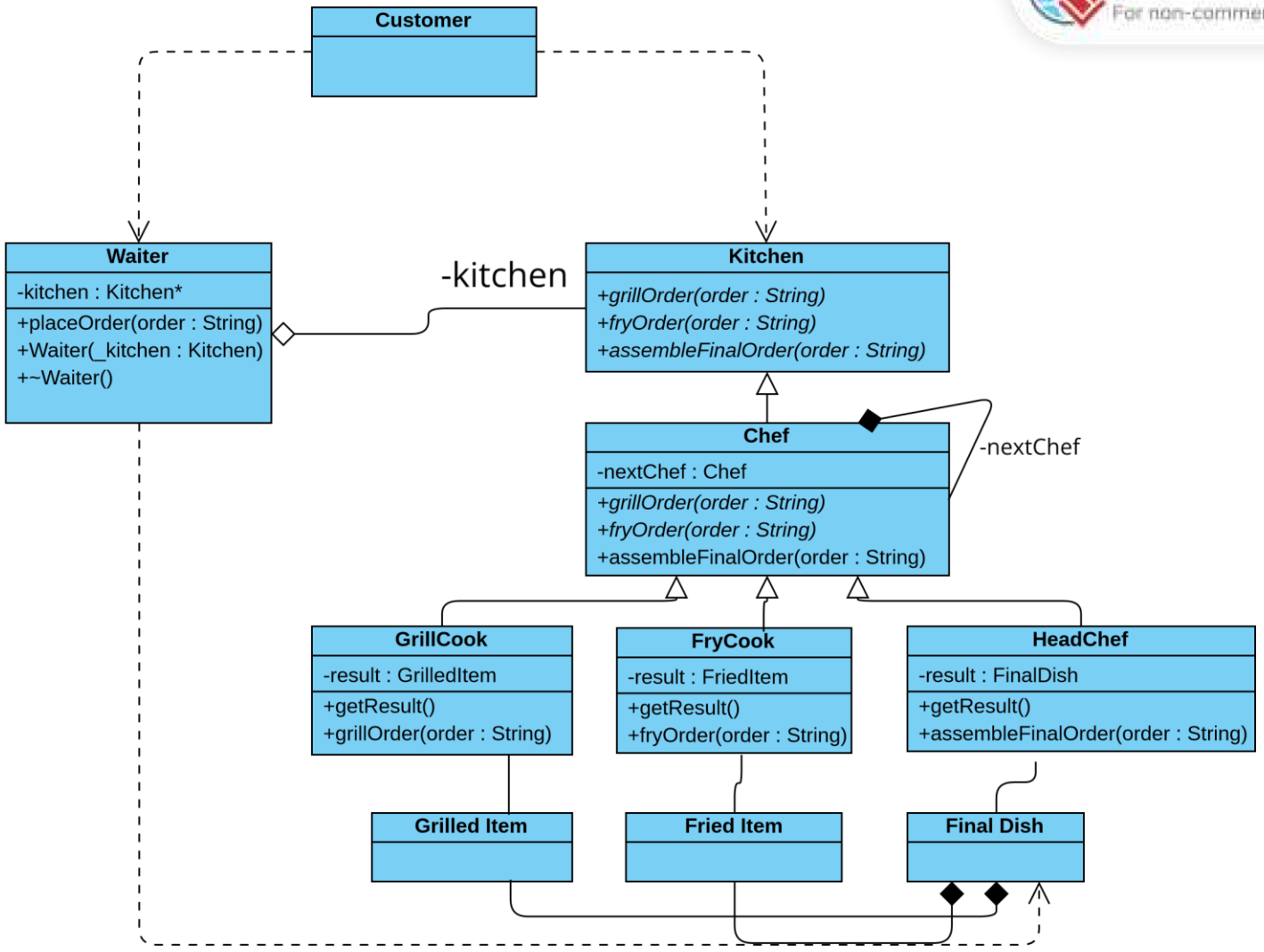
Mediator UML Class Diagram:



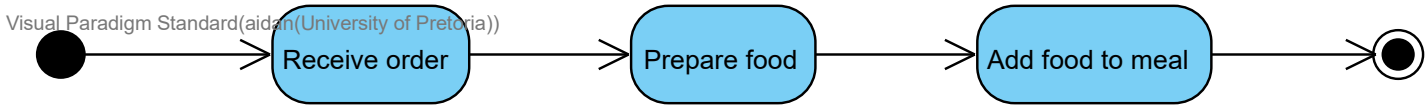
Chain of Responsibilty UML Class Diagram:



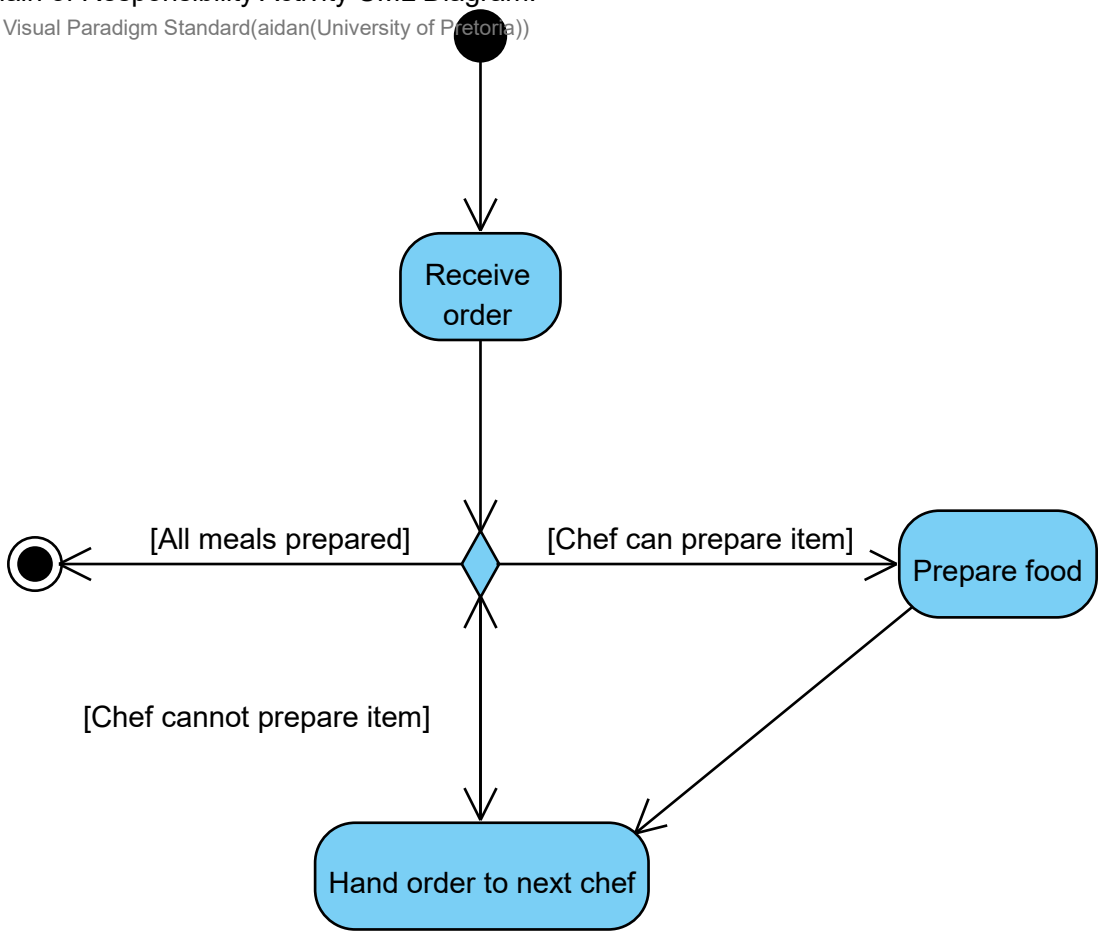
Builder UML Class Diagram:



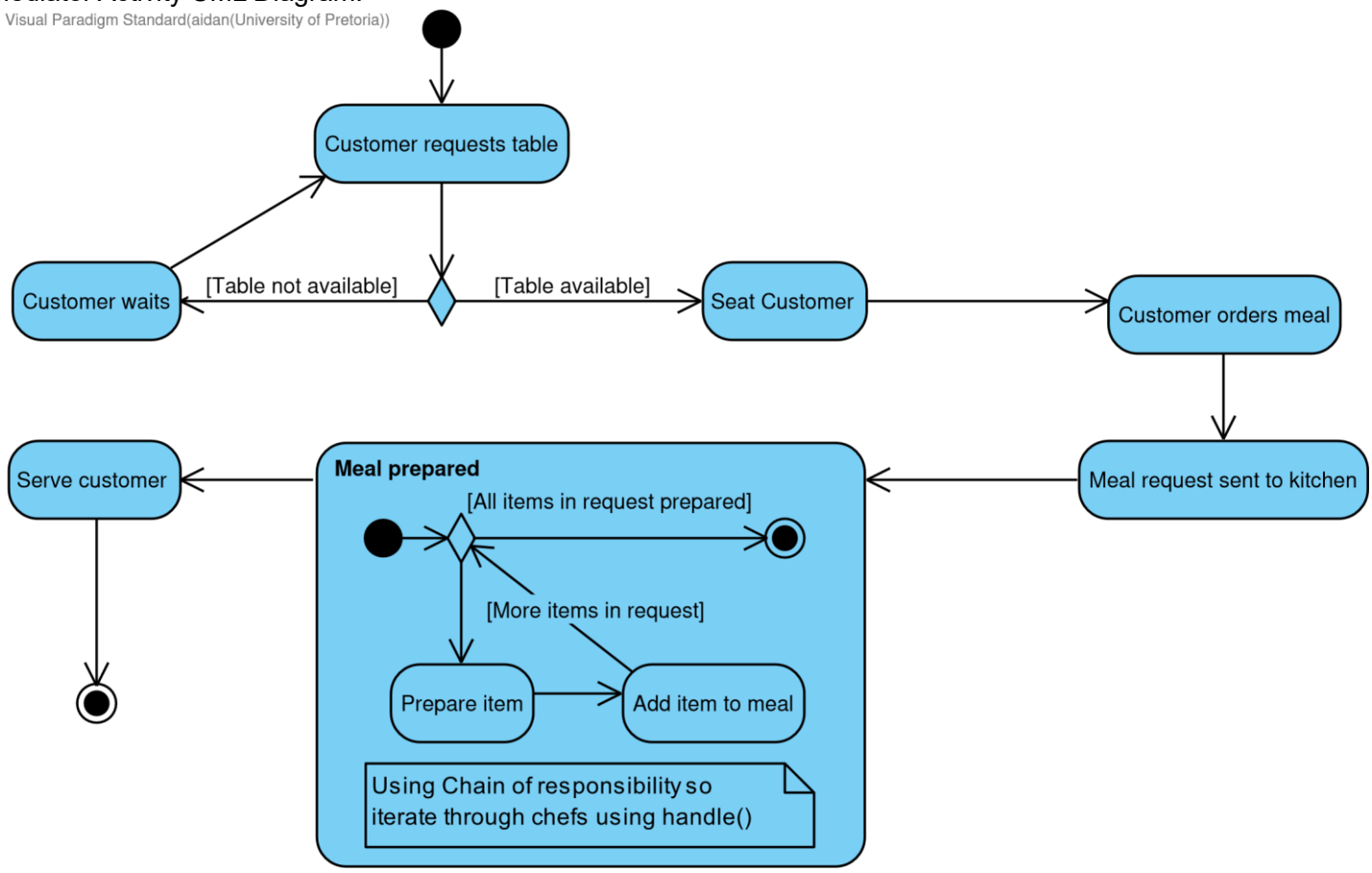
Builder Activity UML Diagram:



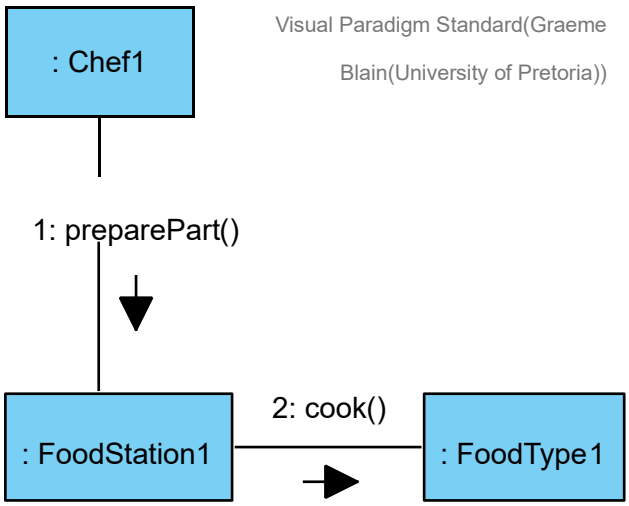
Chain of Responsibility Activity UML Diagram:



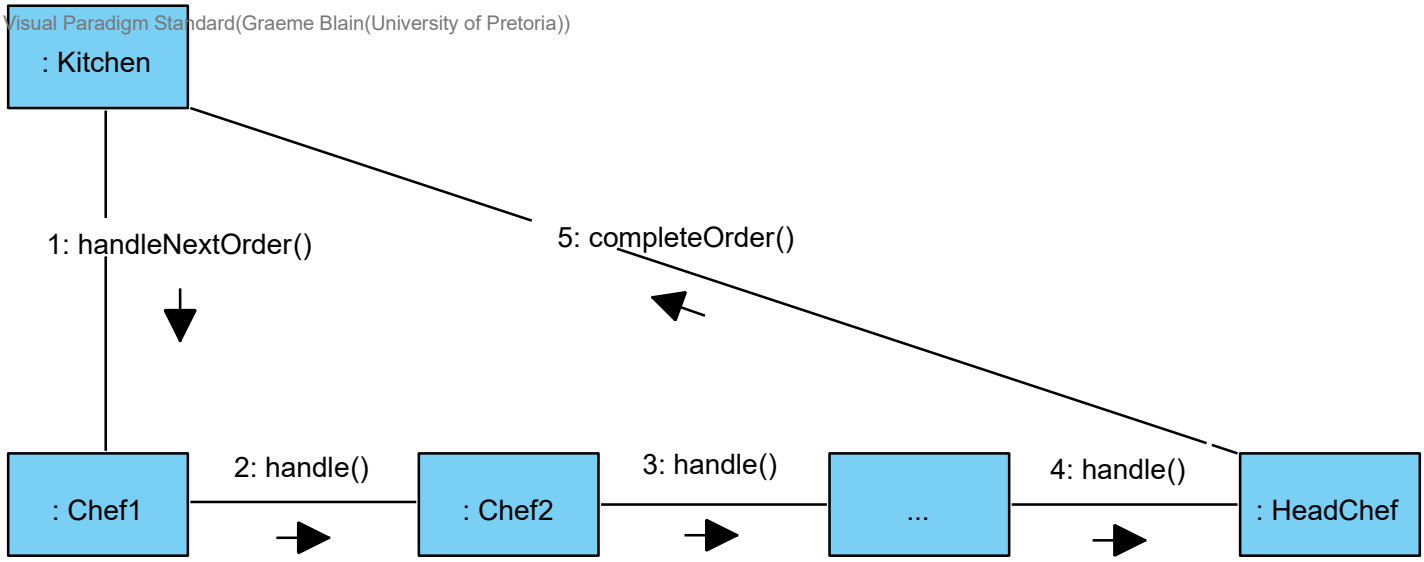
Mediator Activity UML Diagram:



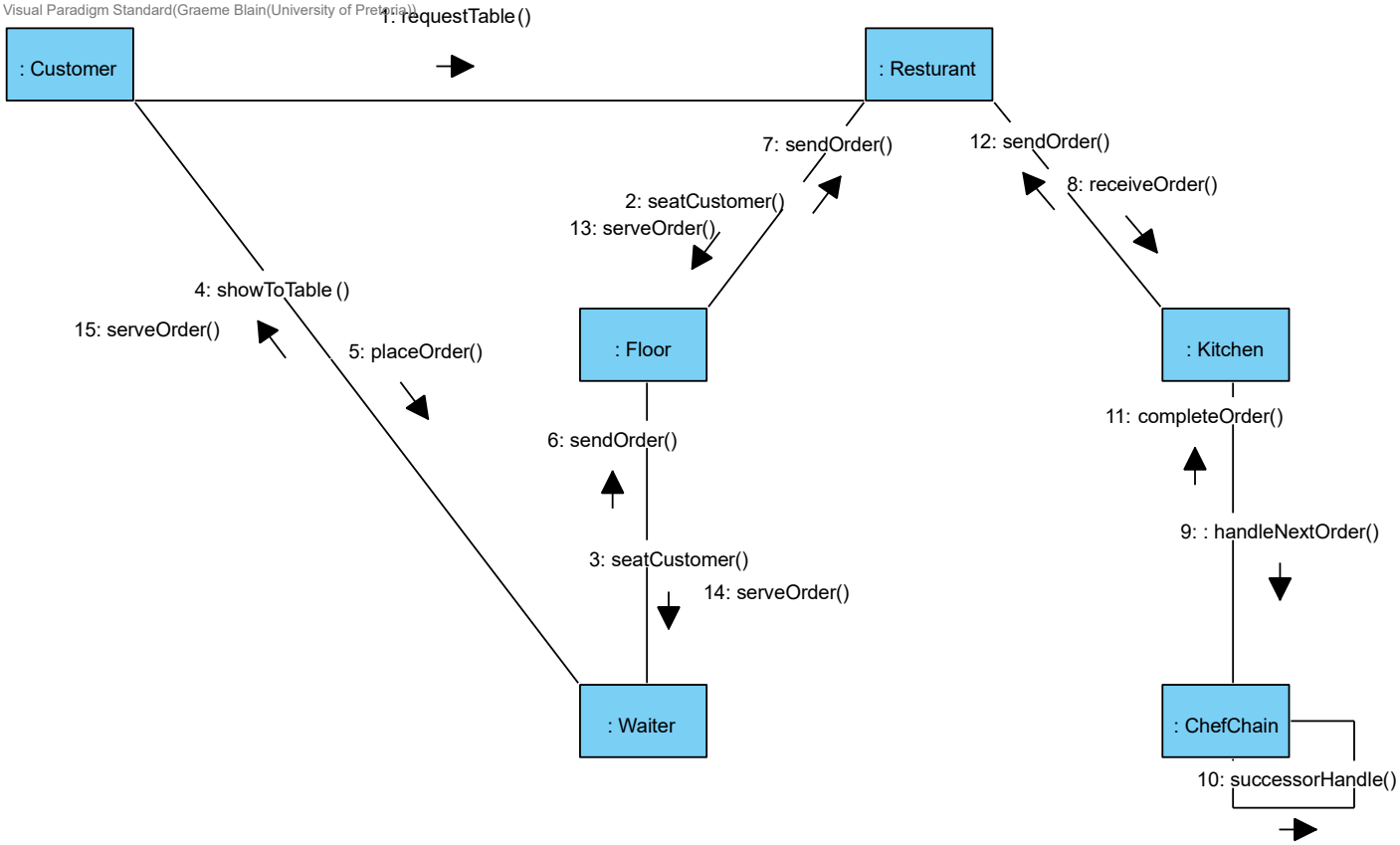
Builder Communication UML Diagram:



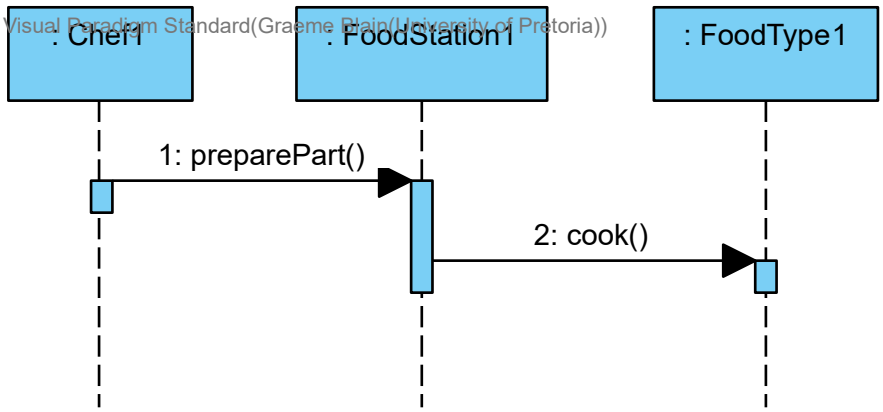
Chain of Responsibility Communication UML Diagram:



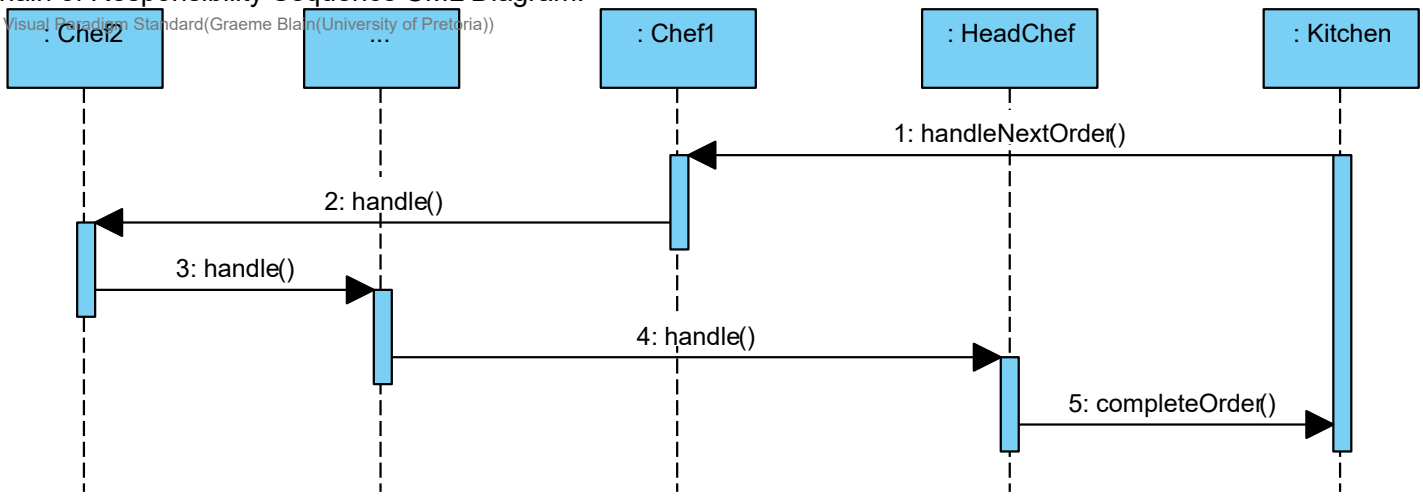
Mediator Communication UML Diagram:



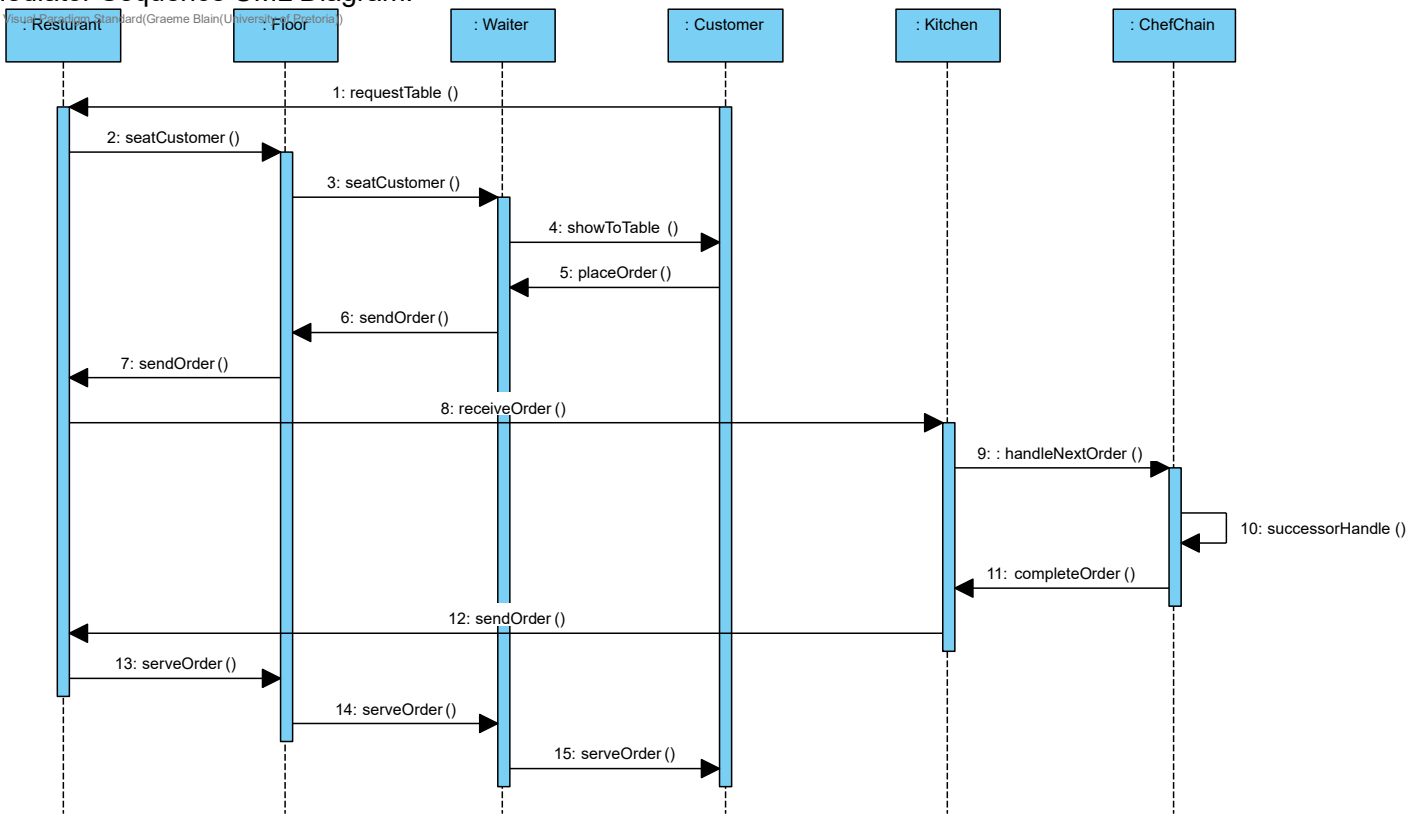
Builder Sequence UML Diagram:



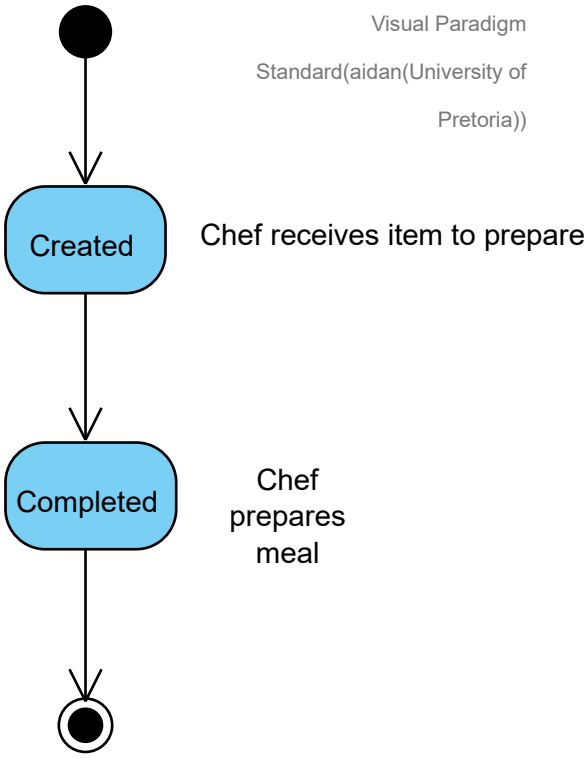
Chain of Responsibility Sequence UML Diagram:



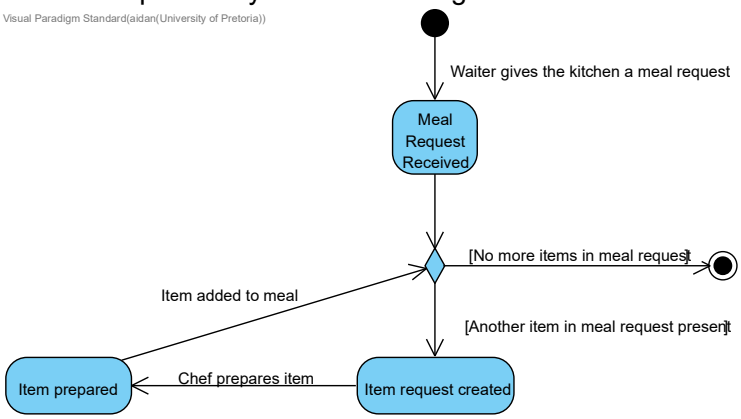
Mediator Sequence UML Diagram:



Builder State UML Diagram:

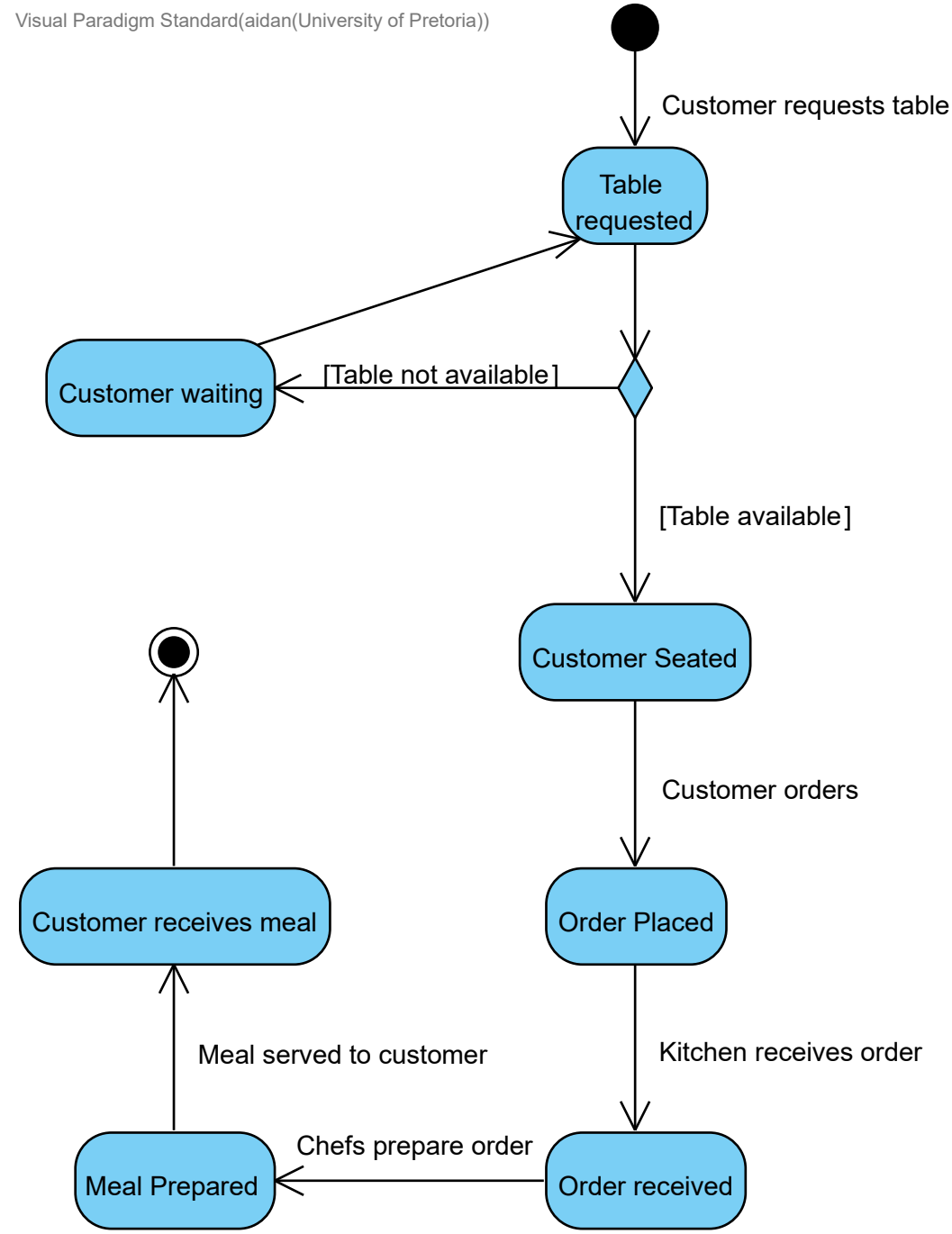


Chain of Responsibility State UML Diagram:



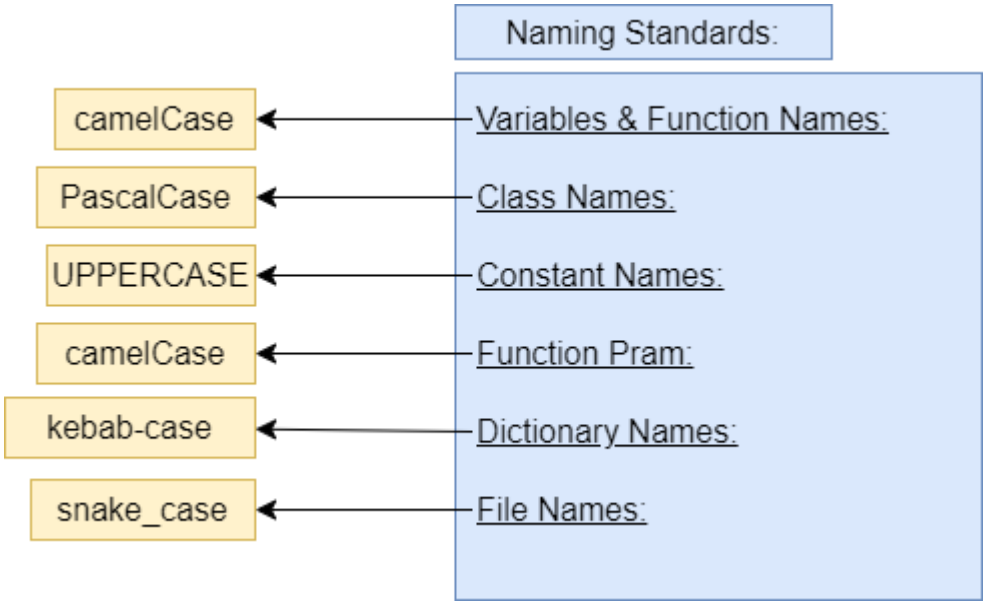
As Chain is being used, the loop will not be explicit:
The entire meal request will be handed over to the kitchen and the handle() function of each chef will be used to iterate through each item requested in the meal (As each chef will be responsible for a specific item, if an item is found in the meal request that corresponds to that chef, it will add that to the meal object that will be returned)

Mediator State UML Diagram:



Coding Standards Diagram

- File informational headers
  - A file informational header will be placed at the beginning of each source file.
  - Format
    - File name
    - Brief description of the purpose of the file
    - Programmers who collaborated on it
    - Revision history notes
- Naming conventions
  - Variable and function names have to be meaningful so that anyone can understand the reason of using it
  - Variables should be named using camel case lettering starting with small letters (e.g localVariable)
  - Constants should use capital letters only (e.g CONSTANTVAR)
  - Avoid the use of digits in variable names
  - Names of functions and function parameters should be written in camel case lettering starting with small letters
  - Name of a function must clearly and briefly describe the reason of using the function
  - File names should be in snake case



- Indentation
  - Use tab indentation
  - Each nested block should be properly indented and spaced
  - All braces should start from a new line and the code following the end of braces should also start from a new line
- Header files
  - Each module has it's own corresponding header file that describes its interface. (i.e no mixing of interfaces into one header file)
  - Header files will not contain any code, just declarations

    - Header files will not be used to initialise data
- Subroutine headers
    - Each subroutine in the file should have a header before the beginning of its declaration
    - Format
        - Name
        - Brief description of its purpose
        - Subroutines called in support of this one
        - Return (if applicable)
        - Programmers
        - Date
- Comments
    - All comments should comply to Doxygen commenting syntax
    - Comments should be presented in paragraph form prior to a block of code
    - Indent block comments on the same level as the block being described
    - Although block comments will be the primary form of commenting, use inline comments where appropriate
    - An inline comment shall accompany each definition of a variable
- Git standards
    - No pushing directly to the master branch
    - Everyone branches off of the master branch then code will be merged into the master branch at a later stage
    - Use clear and concise commit messages