COS 214 Project

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

- Date Issued: **10 October 2023**
- Date Due: **7 November 2023** at **11:00am**
- Submission Procedure: **Upload to ClickUP**
- Submission Format: **archive (zip or tar.gz)**

1 Introduction

1.1 Objectives

In this Assignment you will:

- Design a system using patterns of your choosing to address design issues.
- Implement a system that has been designed with proper use of patterns.
- Demonstrate your understanding of the various patterns discussed during the semester.

1.2 Outcomes

When you have completed this Assignment you should:

- Have demonstrably mastered the application of patterns
- Be able to design systems with these design patterns to meet various quality constraints and design principles.

2 Constraints

1. This assignment must be completed in teams of 5 – 7.
2. You may ask the Tutor for help but they will not be allowed to give you the solutions.
3. You must demo the system at the conclusion of the assignment and all participants must be present at the demonstration.
4. You are required to use doxygen to document your implementation.
5. You are required to use Git to cooperate with other team members.
6. Rough estimate of timeline:
    - 13 October: Form Group and discuss initial design.
    - 24 October: Complete prac 5 portion of design and start implementation.
    - 31 October: have a Completed design and have three quarters of the implementation done, also start demo prep.
    - 5 November: Complete implementation and continue with Demo prep.
    - 7 November: Demonstrate system

3 Submission Instructions

You are required to upload all your source files (that is `.h` and `.cpp`), your Makefiles, and a single PDF document and any data files you may have created, in a single archive to ClickUP before the deadline. A demonstration of the implementation will be conducted after submission. It is required to create a main that allows for the proper

demonstration of the functionality provided by the system and documentation to highight how design patterns were applied to provide this functionality.

4 Mark Allocation

Task	**Marks**
Practical assignment 5	0
Design	25
Implementation	35
Report	20
Demonstration	20
Bonus marks	10
TOTAL	110

5 Assignment Instructions

This assignment will require you to design a restaurant simulator. A fair bit of the assignment is up to the team to design including the implementation as a pure simulation or as a "restaurant tycoon" game.

A restaurant is a chaotic ensemble of various processes happening at the same time (Multithreading is not required) to result in the production and sale of food to customers. These processes can be small, from being seated at a table, to massive processes such as the cooking of food at various workstations and being passed between chefs.

For the purpose of this assignment we will be dealing primarily with two distinct areas, the floor and the kitchen, of the restaurant and how they must communicate.

5.1 The floor

The floor will be where customers are seated and managed. Here we will have some system for a customer to request to be seated and if available given a table, this system mimics a Maitre D showing you to the table. Thereafter a waiter will come over and take your order. If the customer is not ready to order they may ask the waiter to come back after a while. When a customer places an order the waiter must pass it to the kitchen and then when the kitchen has prepared the order the waiter must return the order to the table.

At the end of the dinner a bill must be presented to the table and by request a bill may be split into sub-bills so that more than one payment may be made from various parties at the table.

Tables will also be managed by floor staff as tables can combined or split to be able to fit various sized parties. The restaurant may have a booking system or walk in based service. Every waiter will have tables assigned to them that they will take responsibility for.

Importantly customer expectations need to be managed. If the customer is happy, they may tip more. However if we make a customer unhappy they might make a complaint.

The restaurant is a "Build-Your-Own-" restaurant meaning a customer will have the ability to create an order out of a list of available options and may ask for specific methods of preparation (grilled vs fried for example).

Seeing as we want to present as a classy establishment we may allow a customer to start a Tab where they will only pay for a bill at a later date.

5.2 The kitchen

The kitchen is where most of the chaos and inter operation comes into effect.

An order must be passed by a waiter to the kitchen who will then take orders as they come and start producing them. Different chefs will be responsible for different parts of the preparation process, for example the head chef finishes plates while the fry cook will always work at the fryer.

A dish may be passed between various stations before returning to the head chef to complete the plating before it can be sent to a customer via the waiter.

Whenever an order is completed the kitchen will notify the waiter that they can come and collect the order to take it to the table

5.3 Additional instructions

Chefs as well as waiters and managers may do rounds and visit tables to ensure that the customer is happy and good service has been provided.

If you have not identified enough distinct patterns in the design above you may choose to extend the design such that you do. Some ideas include:

- Bar with various cocktails
- Valet service
- Inventory and accounting, useful if you want to go the tycoon route.

6 Tasks

Task 1: Practical assignment 5 ..... (0 marks)

You will be required to submit an initial view of your design outlined in the next section as practical 5 as a PDF file (remember the discussion in class).

1.1 Submit partial UML class diagrams for at least 3 patterns as they will be used in the system.

1.2 For every other UML diagram(activity, state, etc.) specified provide at least one diagram in the submission.

1.3 Research and write-up a coding standards diagram, including naming standards and git standards.

Task 2: Design ..... (25 marks)

2.1 Identify the requirements of the system and document it in your report

2.2 Using the requirements identified, model processes using activity diagrams.

2.3 Identify patterns that will complement the processes and requirements identified.

2.4 A minimum of 10 distinct design patterns are required. You will be heavily penalised for not including at least 10 patterns.

2.5 Draft a complete UML class diagram for the entire project. Remember a class may be a participant in multiple patterns simultaneously.

2.6 Draft sequence and communication diagrams that show message passing in the system

2.7 Provide a state diagram showing how an object might change state as time passes in the system.

2.8 Provide object diagrams showing the state of objects in the system as it is running.

Task 3: Implementation ..... (35 marks)

3.1 Implement the design from the previous section using at least a text based interface.

3.2 The use of a GUI is not a requirement but will gain extra marks.

3.3 Consideration may be given to implementing your coding standards in your linter for bonus marks.

3.4 The use of git and GitHub is a requirement and every member is expected to make at least 10 significant commits to the project.

3.5 Your code must be documented using doxygen.

3.6 You are required to set up unit testing and every member must have written unit tests. Full coverage is not a requirement.

3.7 Integrating your unit tests with github jobs is not required but would be considered for extra marks.

Task 4: Report ..... (20 marks)

As part of your assignment you are required to set a PDF document that includes the following:

4.1 A write-up of all the research you have done to complete this assignment including references.

4.2 Your reasoning for the design decisions you have made.

4.3 A writeup showing how all the patterns have been used, also note what problem the pattern solved in your implementation.

4.4 Any and all assumptions, alterations, and design decisions.

4.5 UML diagrams in support of the text presented in the report.

Task 5: Demonstration ..... (20 marks)

The entire team must be present at the demonstration. You will be critiqued on your demo preparedness and presentation skills.

Task 6: Bonus marks ..... (10 marks)

6.1 Bonus marks for GUI (5)

6.2 Bonus marks for Dev Ops(linter and CI/CD) (5)