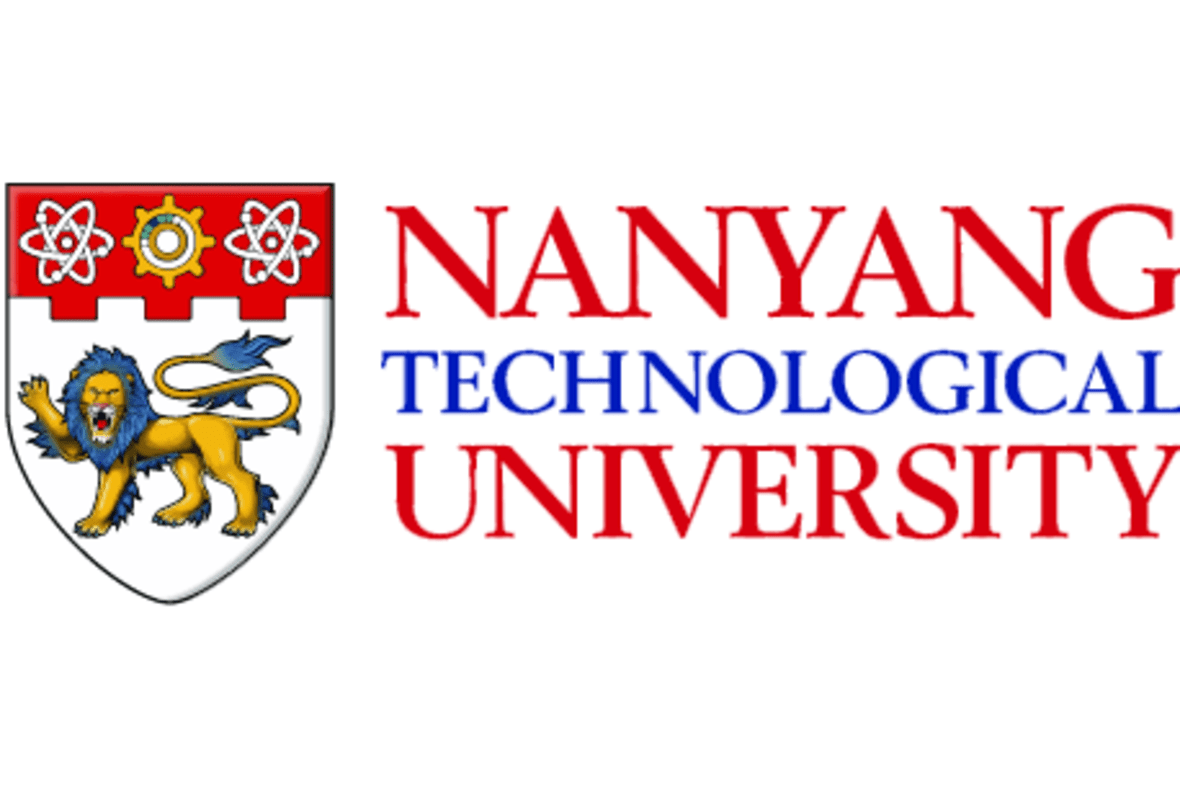
**SCHOOL OF Computer Science and Engineering**



**CZ2002**

**Objected-Oriented Design & Programming**

**Assignment**

**Written by:**

**-----  
-----  
Lee Shan Chao Charles  
-----**

Table of Contents

[**1.** **Declaration of Original Work for CE2002/CZ2002 Assignment** 3](#_Toc528499161)

[**2.** **Design Consideration** 4](#_Toc528499163)

[**3.** **Detailed UML Class Diagram** 5](#_Toc528499171)

[**4.** **Detailed UML Sequence Diagram of stated function** 6](#_Toc528499177)

[**5.** **Testing** 7](#_Toc528499181)

[**References:** 10](#_Toc528499188)

# **Declaration of Original Work for CE/CZ2002 Assignment**

Declaration of Original Work for CE/CZ2002 Assignment We hereby declare that the attached group assignment has been researched, undertaken, completed and submitted as a collective effort by the group members listed below. We have honored the principles of academic integrity and have upheld Student Code of Academic Conduct in the completion of this work. We understand that if plagiarism is found in the assignment, then lower marks or no marks will be awarded for the assessed work. In addition, disciplinary actions may be taken.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Course | Lab Group | Signature/Date |
|  |  |  |  |
|  |  |  |  |
| Lee Shan Chao Charles | CZ2002 | 3 | 21 Apr 2020 |
|  |  |  |  |

1. **Design Consideration**

**Design Patterns:**

Considering our application is used by hotel stuff in different scenarios such as making reservations, creating orders, etc. An abstract boundary class is created and different boundary classes are extended from it for different use cases.

Factory Design Pattern is implemented in the top layer of our system to return a selected boundary class based on user input.

The benefits and OO concepts we implemented Factory Design Pattern are:

1. Use polymorphism for boundary object creation and encapsulate the boundary creation from clients.
2. Easy to extend, which satisfies the open-closed principle. For example, we can extend our application to support checking employee information of the hotel by adding the Employee Boundary with the corresponding entity and control classes without affecting other objects and classes.
3. Support lazy initialization and singleton pattern, which only one boundary can be created in run time the created boundary object is based on the user input.

From entity layer to boundary layer, Entity Control Boundary (ECB) pattern is implemented. In our application, one type of entity class has a corresponding control and boundary class, and all the interactions between the boundary class and entity class are processed in control classes.

The benefits we implemented ECB Pattern are:

1. One layer only can interact with the next layer, which means that users only interact with the boundary classes, boundary classes only interact with control classes and entity classes only interact with control classes. Separate the top layer from the bottom layer.
2. A specified boundary and control class are created for a corresponding user case. When modify a user case, it won’t affect the functionalities of other user cases in the system.

**Assumptions:**

# **Detailed UML Class Diagram**

1. **Detailed UML Sequence Diagram of stated function**

.

# **Testing**

Guest Scenario:

1. Create Guest
2. Update Guest
3. Search Guest

Room Scenario:

1. Create Room
2. Update Room
3. Search Room

Reservation Scenario:

1. Create Reservation
2. Update Reservation
3. Delete Reservation
4. Search Reservation

Check-In Scenario

1. Walk in
2. By Reservation

Room Service Scenario:

1.Create Order

2.Update Order

3.Search Order

1.Display Menu

2.Update Menu

Check-out Scenario: