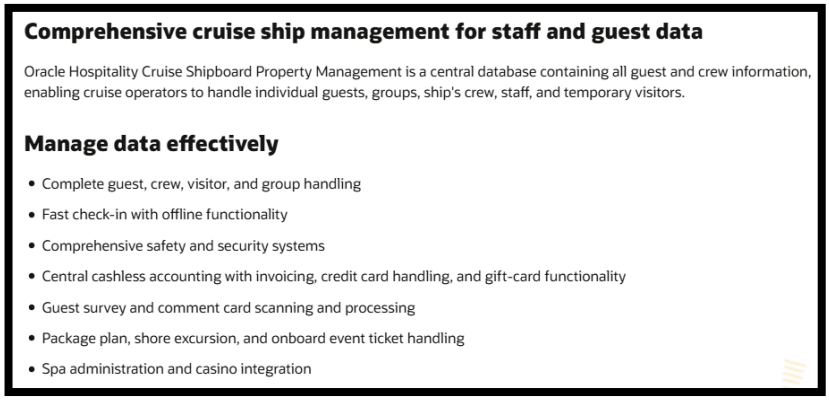
Unit 12 Seminar

Design Patterns in a Python Program

Question 1

Define the test cases for the following system development: Oracle. (n.d.) Cruise Ship Management and Cruise Software.

Oracle (no date) *Cruise Ship Management and Cruise Software*. Available at: https://www.oracle.com/uk/hospitality/cruise/ (Accessed: January 23, 2025).



Test Case for Guest and Crew Data Management

Test Case ID: TC001
Test Case Name: Verify Management of Guest and Crew Data
Objective: Ensure that guest, crew, visitor, and group information is managed and stored accurately in the central database.
Preconditions: User must be logged in with appropriate permissions to manage data.
Steps:

1. Access the guest and crew data management section.
2. Add a new guest.
3. Add a new crew member.
4. Add a temporary visitor and group data.
5. Modify existing guest and crew details.
6. Search for a guest, crew member or visitor by their name or ID.
7. Verify that all data is saved correctly and accessible for updates.

Expected Result:

- Data for guests, crew, visitors and groups is saved and updated accurately.
- No data loss occurs when updates are made.
- Data is searchable by relevant fields.

Question 2

Which design patterns do you consider to be compatible with others, and why?

TutorialsPoint. (no date) *Design Patterns in Java Tutorial*. Available at: https://www.tutorialspoint.com/design_pattern/index.htm (Accessed: January 23, 2025).

Factory Method

The Factory Method is a creational design pattern that defines an interface for creating objects but allows subclasses to alter the type of objects that will be created.

Abstract Factory

The Abstract Factory is a higher level pattern that provides an interface for creating families of related or dependent objects without specifying their concrete classes.

These two patterns are often used together to create a flexible and modular design. While the Abstract Factory provides a family of related products, the Factory Method is used within each concrete factory to determine the specific product to create.

Question 3

Read Zhang & Budgen (2012). Which design patterns are used most commonly, and why?

Zhang, C. and Budgen, D. (2012) “What Do We Know about the Effectiveness of Software Design Patterns?,” *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, 38,(5). Available at: https://ieeexplore-ieee-org.uniessexlib.idm.oclc.org/stamp/stamp.jsp?tp=&arnumber=5975176.

In the study by Zhang and Budgen (2012), they investigated the empirical effectiveness of several design patterns. The patterns they examined included:

- Composite
- Observer
- Visitor
- Singleton
- Factory Method
- Abstract Factory
- Builder
- Prototype
- Adapter
- Facade
- Decorator

They found that the Composite and Observer design patterns were regarded as the most effective and generally useful. The Composite pattern was found useful for its ability to simplify tree like structures and work well when managing hierarchical relationships between objects. The Observer pattern was also mentioned for its ability to promote loose coupling in systems by allowing one object to notify other objects of changes without knowing who or how many observers there are.