**Sequence Diagram Lab:  
Composition, Generalization, and Service Controllers**

Objectives:

* Create class and sequence diagrams with generalized and specialized objects
* Create a class and sequence diagrams for objects in a composition relationship
* Use Service controllers to show interfaces with subsystems.

This is an individual lab. Complete a Visual Paradigm model as per the following requirements. Copy your diagrams to a MS-Word compatible file and include both the Word and source .VPP files in your submission.

**Case Study:**

Relaxing Stay Hotels is a hotel chain throughout Ontario. They own multiple hotels throughout the province. Each hotel has a manager assigned to it, who is responsible for hiring staff, dealing with guests’ issues, and so forth. There are some hotels that are close to each other, and it’s possible for one manager to manage two hotels. There are also cleaning staff and front desk staff, who are assigned weekly work shifts. The cleaning staff and front desk staff are paid an hourly wage, while managers are salaried employees. Legally, all managers must have a yearly certification, that indicates that they know the rules and legal issues regarding their job.

The company needs a system to help manage their employees and rooms.

**Part 1: Class Diagram**

Create a Class Diagram based on the case study background above and the sequence diagrams below. Demonstrate that you understand Composition and Inheritance / Generalization, where applicable.

A computer screen shot of a diagram

Description automatically generated

A screenshot of a computer

Description automatically generated

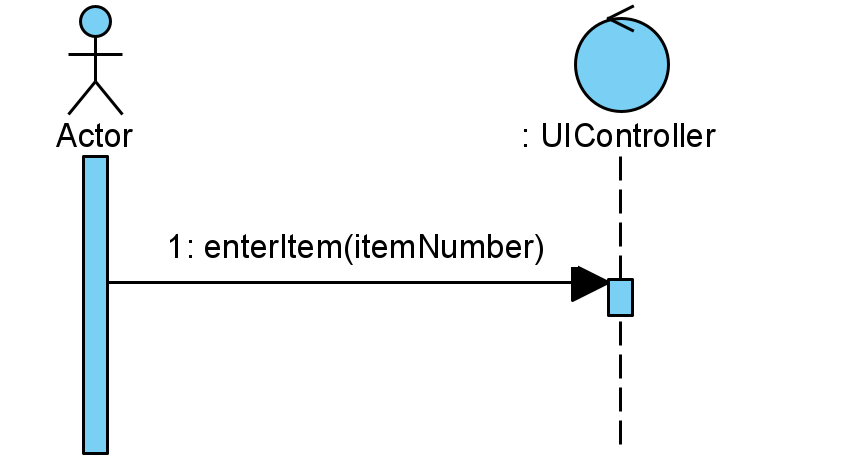
**Part 2: Sequence Diagrams**

Create a Sequence Diagram for each of the sequence diagrams below. Demonstrate that you understand Encapsulation / Composition, Inheritance, and the appropriate use of Service Controllers, where applicable.

**Part 3: Operations**

Each message in the sequence diagram is an operation assigned to your classes (including controllers). **Update each class with it’s operations.** Remember, returns are not operations. Operations in your classes do not include parameters.

For example:

The UIController has the operation: *enterItem*.

*enterItem* ends at the UIController lifeline and therefore belongs to the UIController.

The Customer calls the *enterItem* operation, which is therefore an operation of the UIController.

To add operations to classes in Visual Paradigm, right click on the class in the class diagram and select *Add*, then *Operation.*

**Scenarios:**

*Scenario 1: Generate Employee List*

The Office Administrator wants to see a list of all employees

|  |  |
| --- | --- |
| **Office Administrator** | **System** |
| Requests Employee Report | Generates a report showing all employees (name and number of each employee) and the position of each employee (position name) |

*A screenshot of a computer

Description automatically generated*

*Scenario 2: Hire Manager*

The Office Administrator is logged in; they have chosen to add a new Manager after a successful interview process. The new Manager has provided their most recent certification document as either a PDF or scanned JPEG.

|  |  |
| --- | --- |
| **Office Administrator** | **System** |
| Enters manager’s full name and salary. | Create a new manager with the provided name  Generate and assign an employee number |
| Requests to see all hotels | Retrieve and display a list of all hotels |
| Choose a hotel | Assign the hotel to the Manager and set the Manager to the hotel. Display Hotel |
| Upload certification document | Add the certification to the Manager |
| Save | Save the new Manager to the system. |

A diagram of a project

Description automatically generated

*Scenario 3: Create a Service Ticket*

The Manager is logged in to the system, has received a phone call or email from a guest about an issue in a room. Some service issues require more than one cleaning staff to deal with.

|  |  |
| --- | --- |
| **Manager** | **System** |
| Request list of hotels | Retrieve and display a list of all hotels that the user manages. |
| Choose a hotel | Show all rooms for that hotel |
| Select a room number and enter the problem | Create a new service ticket, and assign a ticket number and today’s date to the ticket.  Retrieve and display a list of all cleaning staff. |
| Choose a cleaning staff employee | Add selected cleaning staff to the service ticket. |
| Repeat above step until all cleaning staff are selected |  |
| Save | Save the service ticket. Send an email to each of the assigned cleaning staff with service ticket details |

A diagram of a project

Description automatically generated

*Scenario 4: Set up new hotel*

The company has recently acquired a new hotel, they will need to add it to their system and assign a manager.

|  |  |
| --- | --- |
| **Office Administrator** | **System** |
| Enter hotel’s address | Create a new hotel, with the address provided. Display the hotel information |
| Enter a room’s details, including the number, floor and number of beds, | Add the room to the hotel. Display the room |
| Repeat above step until all rooms have been added |  |
| Request to see a list of all managers | Retrieve and display a list of managers |
| Choose managers | Add selected managers to the hotel  Display hotel information and request to save |
| Save | Save hotel with rooms |

A diagram of a project

Description automatically generated with medium confidence