**Sequence Diagram Lab:  
Encapsulation, 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:**

Top Snazz Apartments is a company that owns apartment buildings throughout the Toronto area. Each building has a property manager assigned to it, who is responsible for renting units, dealing with renter’s issues, and so forth. There are also cleaning staff and a few concierges, who are assigned weekly work shifts. The cleaning staff and concierges are paid an hourly wage, while property managers are salaried employees. Legally, all property 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 rental units.

**Part 1: Class Diagram**

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

Diagram

Description automatically generated

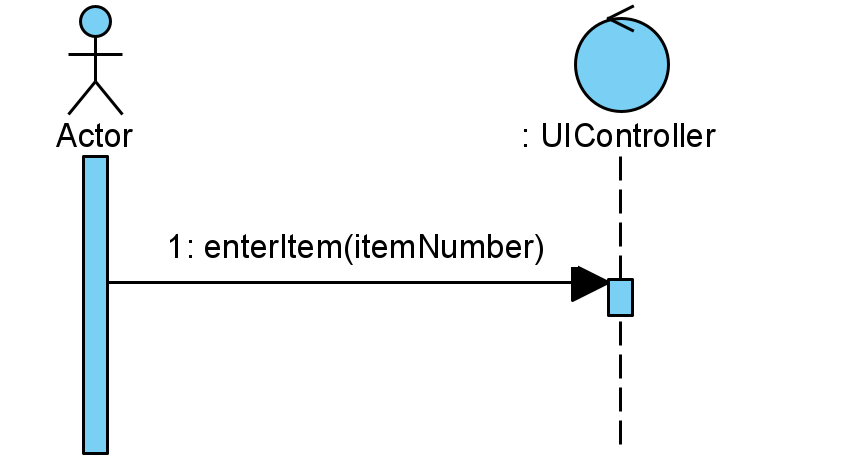
**Part 2: Sequence Diagrams**

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.

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*

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name | Generate Employee List | | |
| Triggering Event | The Office Administrator wants to see a list of all employees – this includes all Cleaning Staff, Concierges, and Property Managers. | | |
| Brief Description | This use case enables the Office Administrator to retrieve a list of all types of employees. | | |
| Actors | Office Administrator | | |
| Related Use Cases |  | | |
| Preconditions | The Office Administrator is logged in and has opened the Employee Menu. Employee options are displayed. | | |
| Post Conditions | The Employee List is displayed | | |
| Flow of activities | Actor | | System |
|  |  | Requests Employee Report | Generates a report showing all employees (name and number of each employee) and the position of each employee (position name) |
| Exception Conditions | Actor chooses to cancel request. | | |

Chart, box and whisker chart

Description automatically generated

*Scenario 2: Hire Property Manager*

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

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name | Hire Property Manager | | |
| Triggering Event | A new Property Manager has been hired after a successful interview process. The new Property Manager has provided their most recent certification document as either a PDF or scanned JPEG. | | |
| Brief Description | This use case enables the Office Administrator to add a new Property Manager. | | |
| Actors | Office Administrator | | |
| Related Use Cases |  | | |
| Preconditions | The Office Administrator is logged in and has opened the Employee Menu and requested to add a Property Manager. | | |
| Post Conditions | Property Manager is added. | | |
| Flow of activities | Actor | | System |
|  |  | Enters property manager’s full name and salary. | Create a new property manager with the provided name  Generate and assign an employee number. Displays new employee |
|  |  | Requests to see all buildings | Retrieve and display a list of all buildings |
|  |  | Chooses a building | Assigns the building to the Property Manager and set the Property Manager to the building. Displays the employee and building assignment. |
|  |  | Save | Save the new Property Manager to the system. |
| Exception Conditions | Actor chooses to cancel request. | | |

Graphical user interface

Description automatically generated with medium confidence

*Scenario 3: Create a Service Ticket*

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

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name | Create Service Ticket | | |
| Triggering Event | Property Manager receives an issue from a resident about a unit. Issue is usually received via phone call or email. | | |
| Brief Description | This use case enables the Property Manager to create a service ticket for an issues. Some service issues require more than one cleaning staff to deal with it. | | |
| Actors | Property Manager | | |
| Related Use Cases |  | | |
| Preconditions | The Property Manager has logged in and has opened the Property Management menu. List of property management options is displayed. | | |
| Post Conditions |  | | |
| Flow of activities | Actor | | System |
|  |  | Request list of buildings | Retrieve and display a list of all buildings that the user manages. Prompts for selection. |
|  |  | Choose a building | Show all units for that buildings. Prompts for selection and problem entry. |
|  |  | Select a unit 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 | Prompts to save. |
|  |  | Save | Assign the service ticket to the unit and save the service ticket and the unit. |
| Exception Conditions | Actor chooses to cancel request. | | |

Diagram, timeline

Description automatically generated

*Scenario 4: Set up new building*

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name | Setup New Building | | |
| Triggering Event | The company has recently acquired a new building, they will need to add it to their system and assign a property manager. | | |
| Brief Description | This use case enables the Office Administrator to add a new building. | | |
| Actors | Office Administrator | | |
| Related Use Cases |  | | |
| Preconditions | The Office Administrator has logged in and requested to add a new building. | | |
| Post Conditions | Building has been setup and added. | | |
| Flow of activities | Actor | | System |
|  |  | Enter building’s address | Create a new building, with the address provided. Prompts for unit details. |
|  |  | Enter a unit’s details, including the number, floor, number of bedrooms, number of bathrooms, and square footage | Add the unit to the building |
|  |  | Repeat above step until all units have been added | Prompts for property managers |
|  |  | Request to see a list of all property managers | Retrieve and display a list of property managers |
|  |  | Choose a property manager | Add selected property manager to the building.  Display building information and request to save. |
|  |  | Save | Save building with units |
| Exception Conditions | Actor chooses to cancel request. | | |

Graphical user interface, application

Description automatically generated

*Scenario 5: Submit Certification*

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name | Submit Certification | | |
| Triggering Event | A Property Manager has just had his Certification updated. The Office Administrator needs to enter it into the system and send a copy of it to the Condominium Authority of Ontario (CAO). | | |
| Brief Description | This use case enables the Office Administrator to add a certification to the Project Manager and send that certification to the Condominium Authority of Ontario. | | |
| Actors | Office Administrator | | |
| Related Use Cases |  | | |
| Preconditions | The Office Administrator is logged in and as list of Property Manager options is displayed | | |
| Post Conditions |  | | |
| Flow of activities | Actor | | System |
|  |  | Request list of all Property Managers | Retrieve and display a list of all Property Managers. |
|  |  | Select a Property Manager | Retrieve and display the selected property manager’s name and employee number, and a list of all their existing certifications. Prompts for additional certifications |
|  |  | Upload a new certification PDF file | Receive the file and add it to the system  Send a copy of the PDF to the CAO.  Display a confirmation message to the user. |
| Exception Conditions | Actor chooses to cancel request. | | |

Graphical user interface

Description automatically generated with medium confidence