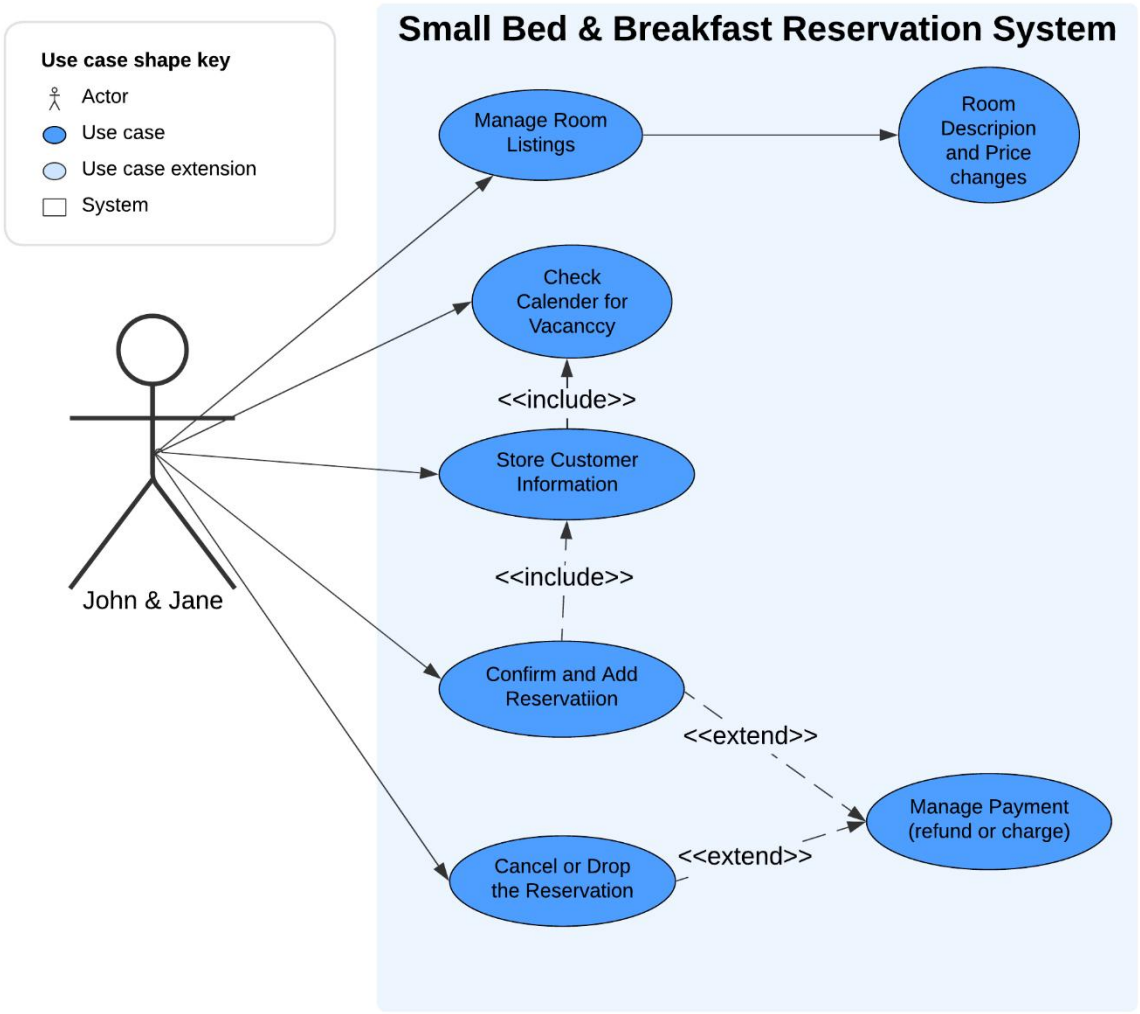
Name: Victoria Lee
Date: 10/29/2024
Class: CMSC 345 7380 Software Engineering Principles and Techniques.
Primary Instructor: John Lee
Week: 2 - Create an Analysis Model for a Small Bed & Breakfast Reservation System (Project1)

1)
Rubric Criteria:
Create UML use case diagram of 3-7 use cases 10%
Your Response: This is the UML use case diagram for the B&B reservation System of John and Jane.



2)
Rubric Criteria:
Write the use case sequence of events for each use case in the use case diagram 20%
Your Response:

Upon receiving a call from a customer, John or Jane will make a room reservation for the customer.

Preconditions:

> If the customer is a regular customer, the customer's information is already in the system including credit card numbers. If the customer is new, add the customer's information to the system after checking the vacancy and informing room listings.

Postconditions:

> 1) a reservation is made for the guest for the requested dates, or
>
> 2) no reservation is made

The sequence of events (scenario):

Use Case 1.0: The Owner Checks Calendar for Vacancy Spots

1. A customer calls the owners for a reservation and specifies the start and end dates needed.
2. The owner looks up rooms available for the date range (start and end) specified in the system.
3. If a vacancy exists, the owner informs the customer of rooms available and their rates from the listings.

Use Case 2.0: The Owner Manages Room Listings

1. After the owner checks the vacancy, the owner checks three types of rooms to the customer and the rates.
2. If available, the owner informs the customer that the first option is a one-room with a private bath and is the most expensive room.
3. If available, the owner informs the customer that the second option is a room with a lake view that is moderately priced.
4. If available, the owner informs the customer that the third option is a normal room and is the least expensive.
5. Then, the owner must wait for the customer's response after informing them of the available types of rooms and the rates from the listing.

Use Case 3.0: The Owner Enters And Stores Customer's Information

1. The customer selects the desired room from the options the owner gave.
2. The owner prompts the customer for information such as the name, address, phone number, reservation dates, room number, room type, and billing information).
3. After the customer decides and responds, the owner enters the payment details into the system.
4. The system creates an invoice for the reservation fee and stores the customer's information.

Use Case 4.0: The Owner Confirms The Reservation

1. The owner sends an invoice for reservation via the system either sent to an email or phone.
2. The customer responds to invoices sent outside of the system.
3. The system notifies of customer payment and a reservation is made with the start and end dates.
4. The system does not notify or block dates and room types for reservations that were specified.
5. The final invoice is prepared where a digital copy is sent (email and phone) while the owner receives a separate copy.

Use Case 4.1: The Payment Is Managed to Charge The Reservation

1. The owner sends an invoice payment charges for a specified reservation selected via the system either sent to an email or phone.
2. The customer provides the payment information if not already inputted through the customer's information or in the system.
3. The system confirms the payment, notifies of customer payment, and a reservation is made with the start and end dates.

Use Case 5.0: The Reservation Is Removed.

1. If the customer calls before the reservation date to cancel, the owner proceeds to cancel the reservation through the system.
    a. The customer calls the owner before the reservation date or time that was set up.
    b. The customer requests a cancellation and provides the reservation information.
    c. The owner finds the reservation in the calendar and deletes the reservation.
    d. The deletion of the reservation alerts the system to send a notification via digital communication methods such as email, phone, or mail.
    e. The room becomes available and open to reservations for the deleted dates or times.
2. If the customer does not call before or fails to pay before the day, the owner may proceed to drop the reservation or charge a late fee through the system.
    a. The owner checks if the initial invoice has not been paid a day before the reservation date.
    b. The owner looks at the calendar, and the system shows that the reservation Boolean (T=Did Pay, F=Did not pay) is false.
    c. The owner selects the reservation and the options of either 1) paying through another method or 2) dropping the reservation.
    d. The owner selects option 2.
    e. The drop alerts the system to send a notification through digital communication methods such as email, phone, or mail.
    f. The room becomes available and open to reservations for the dropped dates or times.

Use Case 5.1: The Payment Is Managed To Charge Or Refund The Reservation.

1. The owner sends an invoice payment charges for a specified reservation selected via the system either sent to an email or phone.
2. The customer provides the payment information if not already inputted through the customer's information or in the system.

3. If the reservation is canceled before the reservation date, a refund is added to the customer's payment.
4. If the reservation has not been paid a day before the reservation date, the reservation is dropped or the owner may charge a late fee to pay through other methods.
5. The system confirms the payment, notifies of customer payment, and a reservation is made with the start and end dates.

Total Use Cases (including the extends): 7

After all the Use Cases are done:

John or Jane asks the guest for their name and completes the room reservation without or without a 1-day guarantee payment as agreed with the guest. Otherwise, if there are no available rooms on the selected date range or the guest does not like any of the available rooms, the use case starts all over again with different start and end dates, or no reservation is made.

3)
Rubric Criteria:
Explain the approach, steps, and rationale of the use case model 25%
Your Response:

Approach:

Firstly, I started off with the actors before the actions taken(tasks). The owners, John and Jane are the actors because they are the only ones with access to the system. The customer is an actor too, but since the customer cannot control the system, the actor's name is just John and Jane, the owners. The customer can provide details that the owner asks for so that the owner can manage the system. So, the actor (owner: John and Jane) on the left specifies that actions were made. Secondly, I created the use case goals that are based on the actors that describe the actions taken to achieve them. I made sure to thoroughly read the scenario and apply it to other template examples online about the UML Use Case Diagrams to help me create it. The approach to creating the use case model for the Bed & Breakfast (B&B) reservation system focuses on the interactions between the owners (John and Jane) and the system. The goal is to design a system that manages reservations, customer information, and payments. Through this approach, I can make sure that it meets the requirements of a UML use case and the diagram as well (Tsui, Karam, & Bernal, 2014).

Steps:

1. The first step is to identify the actors. The primary actor is the owner of the B&B (John or Jane). A customer is not the actor as they can not manage the system, but they can call the owner to change the system (make suggestions through reserving a room, etc). I doubled check with the scenario to make sure that all actors were considered.

2. The second step is to define the Use Cases. The goals of the system are to manage reservations, process payments, and provide information to the owners. I can do this by identifying key

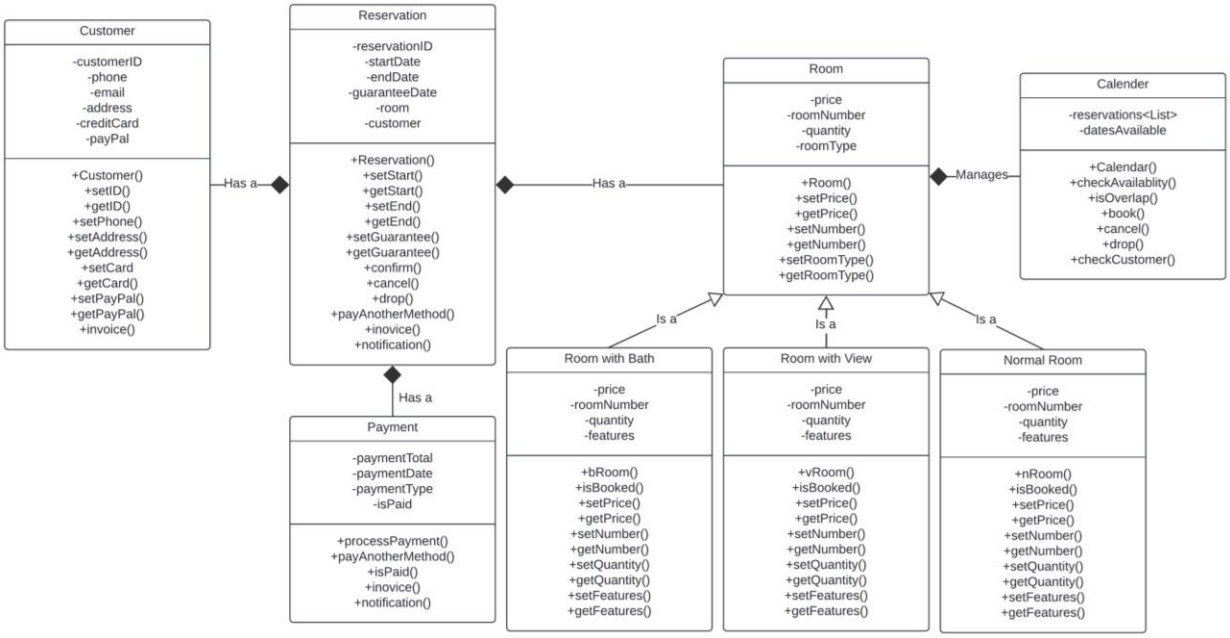
functionalities so that the actors can complete the goals and actions to manage reservations, customer information, and payments. I double-checked this through the scenario and made sure I could write sentences for the use cases.

3. The third step is to outline the details of each Use Case. Based on the goals and actors, the use cases were identified and defined. In other words, organize the use case and make sure you can add more details to it. For instance, start with basic simple sentences about the goals. And then add in more details after building the basis of the use case sentences. To make sure the details are relevant to the scenario, I made sure to reread the scenario and then apply the details.
4. The fourth step is to for each use case, a detailed scenario was created to describe the steps involved in performing the use case. In other words, outline the sequence of events (scenario) to specify the steps involved in the interaction for each use case. Since the previous step was to write it basically, I made sure to go back and embed details into each use case.
5. The last step is to review the use case model before finalizing it. To do this, I reviewed the use case model with the scenario given to ensure that it covers all the functionalities. Once finalized, the resulting use case model provides a representation of the system's functionality and how the owners will interact with it. In other words, it creates the UML use case and the diagram with the scenario paragraphs given.

Rationale:

The use case model captures the essential operations needed for the B&B reservation system, ensuring that John and Jane can manage reservations and payments based on the calendar when the customer calls them. I made sure to utilize the scenario given and appropriately combined the lessons learned to create the actions taken and the steps for each use case. The model is designed with the primary users (John and Jane) in mind, ensuring the system is intuitive. This means that the customer has to guide the owners to achieve the reservation as the owners manage and control the system. The detailed scenarios for each use case help understand the step-by-step process, ensuring no steps are missed. It allowed me to set up the steps and the details for it so that it demonstrates the OO use-case methodology (Tsui, Karam, & Bernal, 2014). The use case model aims to streamline the reservation process, reduce errors, and improve the overall efficiency of the B&B operations. Identifying the actors, setting up a layout for the use case, implementing details for the use case, and then finalizing it allowed me to create this system in a simplistic diagram meeting the requirements. By following this approach, the use case model provides a framework for the B&B reservation system. This can help them have more control in managing and organizing this business.

4)
Rubric Criteria:
Create UML class diagram 10%
Your Response: This is the UML class diagram for the B&B reservation System of John and Jane.



5)
Rubric Criteria:
Explain approach, steps, and rationale of the class diagram model 25%
Your Response:

Approach:

So, the actor (owner: John and Jane) on the left specifies that actions were made. And then the use case goals are based on the actors that describe the actions taken to achieve them. The class diagram has more details for developers and may be somewhat different as it was hard to make it exactly like the Use Case Diagram. I made sure to make it as similar as possible, but if there were some minor changes, it is due to those additional details. I made sure to thoroughly read the scenario and apply it to other template examples online about the UML Use Case Diagrams to help me create it. The approach in creating the UML class diagram for the Bed & Breakfast (B&B) reservation system focuses on modeling the key entities and their relationships within the system. The objective is to provide a clear blueprint of the structure and behavior of the system components, ensuring they work together to support the reservation process (Tsui, Karam, & Bernal, 2014). For this one, the owners can only access and enter information, where the customer can guide the owners so that they can get the reservation and the customer's information into the system. First and foremost, the calendar is the main class of the entire system. Without the calendar, booking the available rooms would be difficult. This means that the calendar manages the system to reserve and book the rooms. The other classes use a has is relationships and room class have an is a relationship. The calendar manages the rooms and the

reservations with the customer. The reservations have a relationship with the customer, payment, and the rooms. The rooms have subclasses with an is a relationship to specify the type of rooms the customer wants that are available according to the calendar. In other words, this setup ensures that the Calendar manages the availability and bookings of Rooms, while Reservations reference to Rooms, Customers, and Payment. Meanwhile, Rooms refer to the subclasses to specify the type of room. Finally, the details are added with the attributes, methods, functions, and relationships.

Steps:

1. The first step was to identify the key entities for each class and the class name. I made sure to determine the main entities involved in the system, such as Customer, Reservation, Room, Room types (3 of them), Payment, and Calendar.
2. The second step was to define the attributes and methods: To do this, I consulted the instructions for each object's attributes. I created the customer's attributes and methods as it was described in the scenario given and then did the same for the rest of the classes. The room reservation was straightforward, and the methods took a while to create to match the previous diagram. The calendar object was easy to create as it is the admin class of all of the rest of the classes. Without the calendar object, there is no reservation to make. For each entity, I made sure to identify the relevant attributes (data) and methods (functions) that describe the entity's state and behavior. The methods and functions were added after the attributes to match with the UML use case model. I tried my best to make the methods and functions as close as possible to the UML use case model to make it accurate.
3. The third step was to establish the relationships. I determined the relationships between entities, such as associations, aggregations, compositions, and generalizations, and could optionally include the multiplicities.
4. Finally, the last step was to finalize the UML Class Diagram. To do this, I made sure to review the diagram to ensure that I accurately represented the requirements based on the scenario given. In addition, I used UML notation to visually represent the classes, attributes, methods, and relationships.

Rationale:

The UML class diagram is different than the UML use case. It allowed me to see the layout of the system's development through attributes, entities, methods/functions, and objects. The class diagram includes all essential entities and their attributes and methods, providing a detailed overview of the system's structure. This allowed me to get a deeper understanding of the system through the classes, modification, and testing. This also allowed me to see how the system functions for this scenario with the B&B reservation system. Since I defined the relationships between entities, the diagram clarifies how different parts of the system interact and depend on each other. Using generalization (inheritance) to model Room types such as Room with Bath, Room with View, and Normal Room, allows for code reusability and a noticeable hierarchy. Although not as simplistic as the UML use case model, this diagram helps create a blueprint of the goals that may or may not work based on the previous model. This can be used in conjunction with the use case diagram for the programming requirements. The class diagram provides the foundation for implementation, making the system easier to develop, scale, and maintain (Tsui, Karam, & Bernal, 2014). The clear definitions of classes and relationships help developers understand the system and make changes when needed (Tsui, Karam, & Bernal, 2014). This is quite

useful for programming and most programmers utilize a UML class diagram or package. By following this approach, the UML class diagram provides a clear and detailed representation of the B&B reservation system and is clear for programmers (developers), ensuring that all components are defined and interact to support the overall functionality (Tsui, Karam, & Bernal, 2014). This helps John and Jane manage their business efficiently for the B&B reservation so that it will be a good reservation experience for their customers.

6)
Rubric Criteria:
Reflect on the learning experience and lessons learned 10%
Your Response:

Creating the UML use case and class diagrams for the Bed & Breakfast (B&B) reservation system involve d understanding the requirements and translating them into structured, visual representations. The proc ess required a detailed analysis of the business needs and a clear understanding of the interactions betw een different system components. I learned that UML use-case diagrams describe how users (actors) interact with the system to accomplish specific goals (GeeksforGeeks, 2024, October). According to Tsui, Karam, & Bernal (2014, Chapter 6), a UML use Case demonstrates Basic functionality, Any precondition for the functionality, Flow of events/scenario for the functionality, Any postcondition for the functionality, and Any error condition and alternative flow. The actor symbol of a human stick figure is just a part of the UML notation. Each actor takes on a certain role with regard to interfacing with the system. This means that each actor is related to a set of use cases. The details of each use case may be further specified in a separate form with additional UML notations. In OO use-case methodology, the steps of identifying the following factors serve as a good requirements analysis methodology for software engineering (Tsui, Karam, & Bernal, 2014, Chapter 6). I also learned that after creating them (UML use case and UML class), these diagrams provide a high-level view of the system's functionality and ensure it meets the requirements. Writing detailed use case scenarios provides a step-by-step guide on how the system should behave, helping to ensure that all functional requirements are met. Through research, some situations to utilize this are when the user wants to comprehend the interactions with the system, the early stages of requirements gathering to identify key functionalities and involving stakeholders in the requirements process (GeeksforGeeks, 2024, October). Moreover, creating the UML class diagram involved identifying the key entities and their attributes and methods (GeeksforGeeks, 2024). This placed the importance of accurately modeling real-world entities to ensure the system's data structure is reflective of actual business processes. I learned that there are a variety of different preparations that go into software development. One of the most valuable lessons learned is the importance of iteration and refinement in the modeling process. There were moments when I made each part either the class or the use case that I realized I needed to improve on and keep on adding additional changes. This made me realize how important planning and formatting is in software development. Establishing relationships between entities helps in understanding how the components of the system interact. It highlighted the importance of correctly using aggregation, composition, and inheritance to represent these relationships. The designing of UML diagrams by creating initial versions, gathering feedback, and refining the diagrams taught me the significance of why software engineering

utilizes them (GeeksforGeeks, 2024). These lessons helped me learn to catch errors early and ensure the final design is efficient. This makes the system easier to scale and maintain through the clear definitions of classes and relationships. It also facilitates future developers to have a better comprehension of the projects when programming them to make sure there are no errors and if there are to debug them efficiently. Overall, creating the UML use case and class diagrams for the B&B reservation helped me reinforce the importance of clear and detailed modeling by understanding the business requirements. It also taught me that system design is scalable, maintainable, and relates to the modern world. This experience is beneficial in future software development projects.

References

GeeksforGeeks. (2024, October). *Use case diagram*. GeeksforGeeks. **https://www.geeksforgeeks.org/use-case-diagram/**

GeeksforGeeks. (2024). *UML class diagram notation*. GeeksforGeeks. **https://www.geeksforgeeks.org/unified-modeling-language-uml-class-diagrams/#uml-class-notation**

Tsui, F., Karam, O., & Bernal, B. (2014). *Essentials of software engineering* (3rd ed.). Jones and Bartlett Learning. https://library-books24x7-com.ezproxy.umgc.edu/toc.aspx?site=VGX8U&bookID=51648