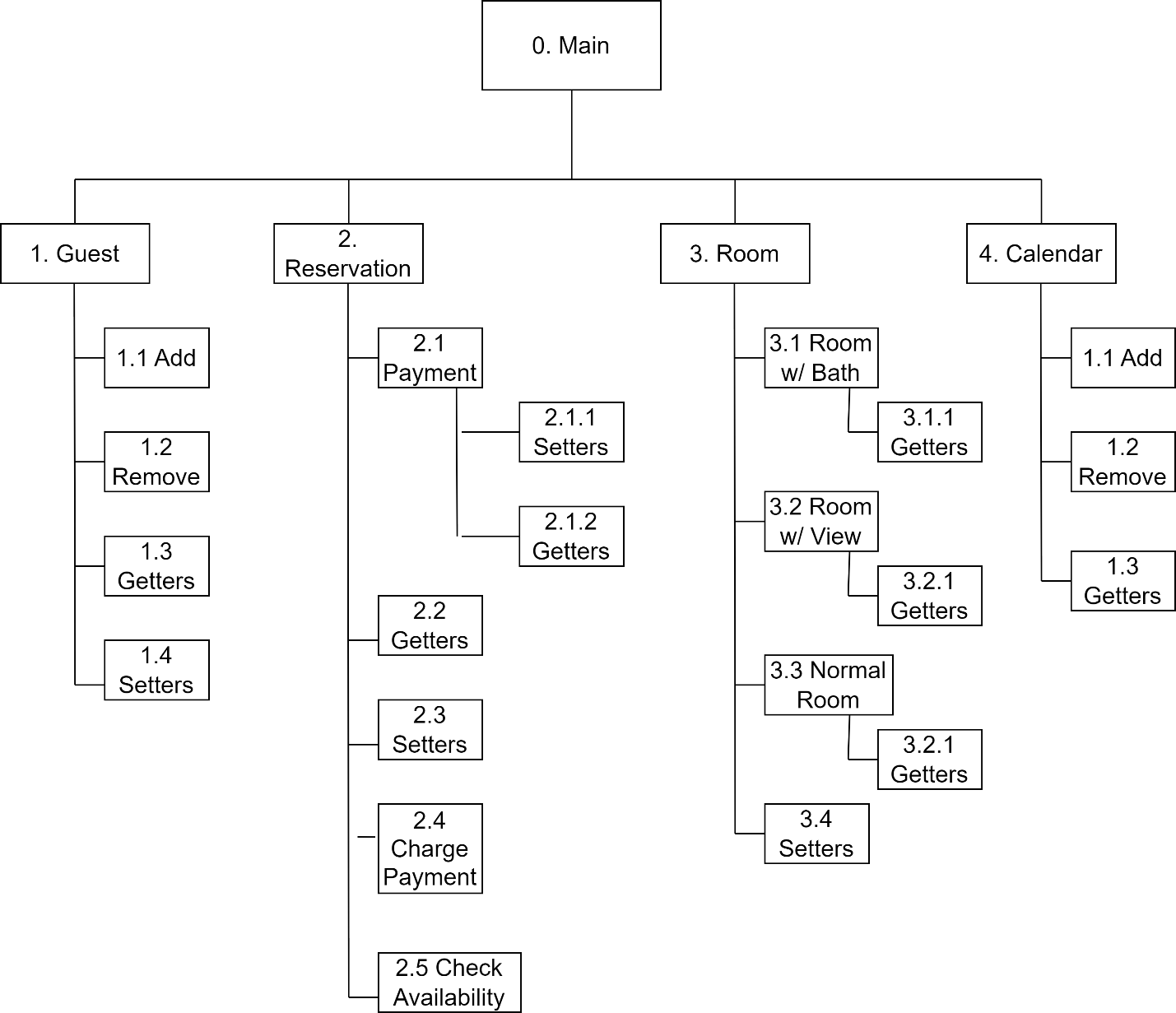
Name: Antonio Scalfar  
Date: 02/06/2024  
Week: 4 - Create a Design Model for a Small Bed & Breakfast Reservation System

1)  
Rubric Criteria:  
Create software system architecture diagram 10%  
Your Response:



2)  
Rubric Criteria:  
Explain approach, steps, and rationale of the software architecture diagram 20%  
Your Response:

My approach to the software architecture diagram was to use a functional decomposition to display the system. I included all of the classes from the analysis class diagram given to us and tried to show classes that were encapsulated within other classes by putting them under the same branches and denoted by the numbering system. For the Guest class, there will be an add, remove, getters, and setters to be able to create or remove a guest in the system and retrieve all their data. For the Reservation class, it will directly use the Payment class, have getters and setters for all information, a check availability method, and a charge payment method. The Payment class will only have getters and setters since the Reservation class will be responsible for the charging payment method. The Room class will be an abstract super class with three classes beneath, Room w/ Bath, Room w/ View, and Normal Room. The Room class will only have setters for choosing the type of Room. Each type of Room will have getters to return the room type and no other methods. The Calendar class will have a getter, add, and remove methods to add or remove a Reservation to a date on the calendar and return a specific date’s current reservations.

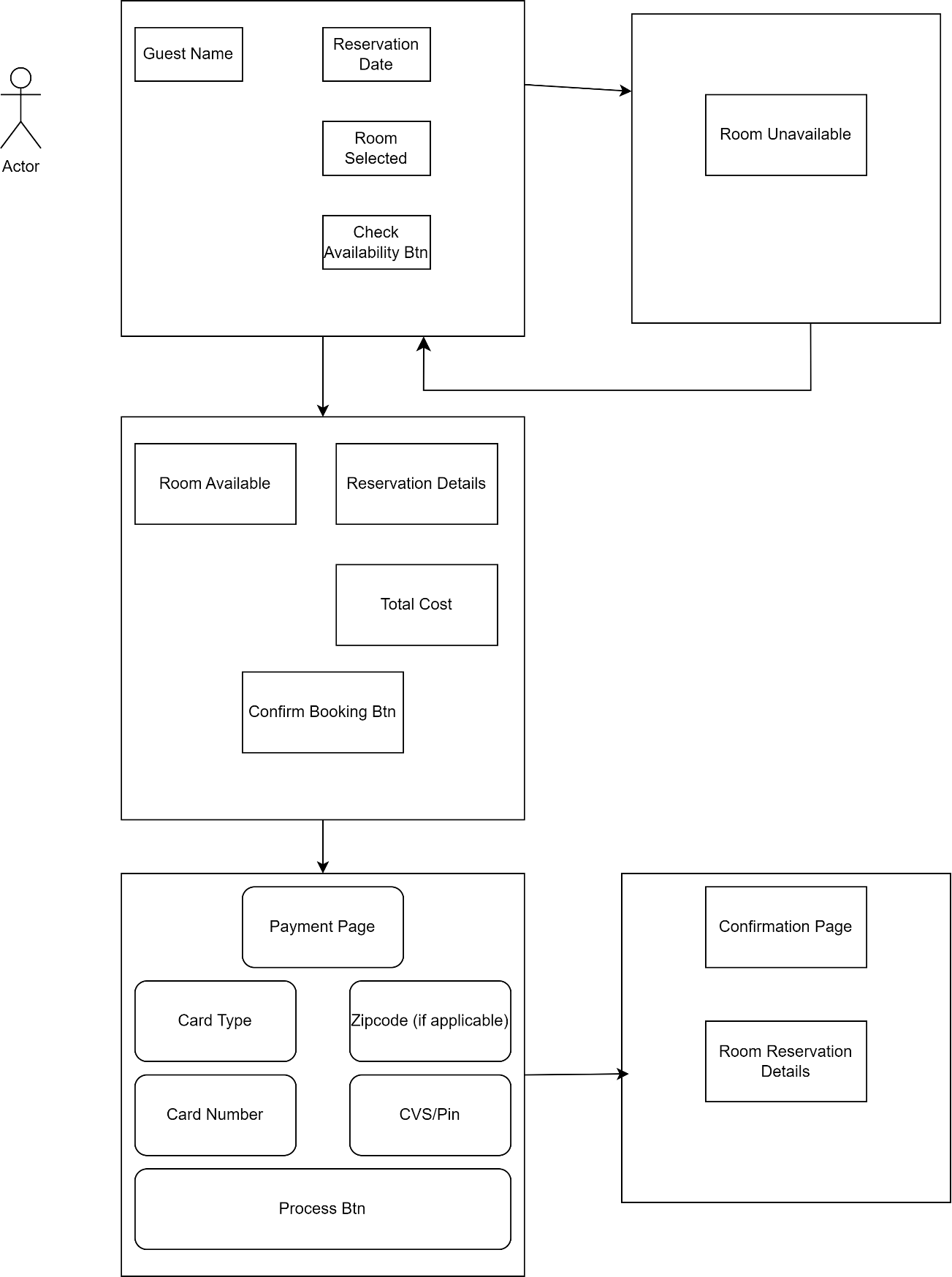
3)  
Rubric Criteria:  
Create detailed UML class diagram 15%  
Your Response:



4)  
Rubric Criteria:  
Explain approach, steps, and rationale of the detailed class diagram model 25%  
Your Response:

To tackle the UML class diagram I first wrote all my thoughts down on paper first, to ensure the majority of the requirements were met with each class and I knew what kind of relationships existed between classes. I then diagramed it out on draw.io, adding each class, its variables, its methods, and its relationship to other classes in the system. I labeled each relationship with either 1 : 1, 1 : n, superclass, or subclass to describe the nature of the relationship. Most of the flow of information is geared towards going to the Reservation class because the Reservation class performs most of the functions of the system and therefore needs the most amount of information stored in the system. The Room class is abstract so that each type of Room can be its own class but has the properties of the Room class itself. I chose to add a guestID to the Guest class to use as a primary key for the class and a foreign key to connect with the Reservation class.

5)  
Rubric Criteria:  
Create user interface mockup 10%  
Your Response:



6)  
Rubric Criteria:  
Explain approach, steps, and rationale of the user interface mockup 10%  
Your Response:

In the user interface mockup, I attempted to give a rough approximation of what John/Jane’s interaction with the system would look like for a typical reservation interaction. I started with the first page containing text fields for guest name, reservation date, and a check availability button. If the date and room are not available, a message page displays this information and reroutes back to the initial page to input new room or date information. If it is available, a page displays the reservation details with the total cost of the reservation and a confirm booking button. Upon confirming the booking, the payment page will appear gathering the payment information for the reservation that will have a process payment button to submit the payment. Lastly, a confirmation page will appear with the reservation details one last time.

7)  
Rubric Criteria:  
Reflect on the learning experience and lessons learned 10%  
Your Response:

This project gave me a great opportunity to strengthen my ability to diagram a system, its class structure, and a user interface for the system. It was, similar to the first project, harder than anticipated and presented unique challenges along the way. I found it difficult to formulate my thoughts for how I wanted the system architecture to look and the interactions/relationships between the classes. I find that writing my thoughts on these topics down on paper and then translating it to a diagram makes it easier to complete. Even with these measures, I still find it difficult to capture the full capabilities of the system and the individual classes. Overall, I can still see flaws in the system I have designed but think it is a great starting point for it. It can be modified in the future but has the basic structure of the system and the class relationships in it. The more diagrams I create, the more comfortable I become with creating them which in turn strengthens the usefulness of the diagrams.