

Project Report

Introduction

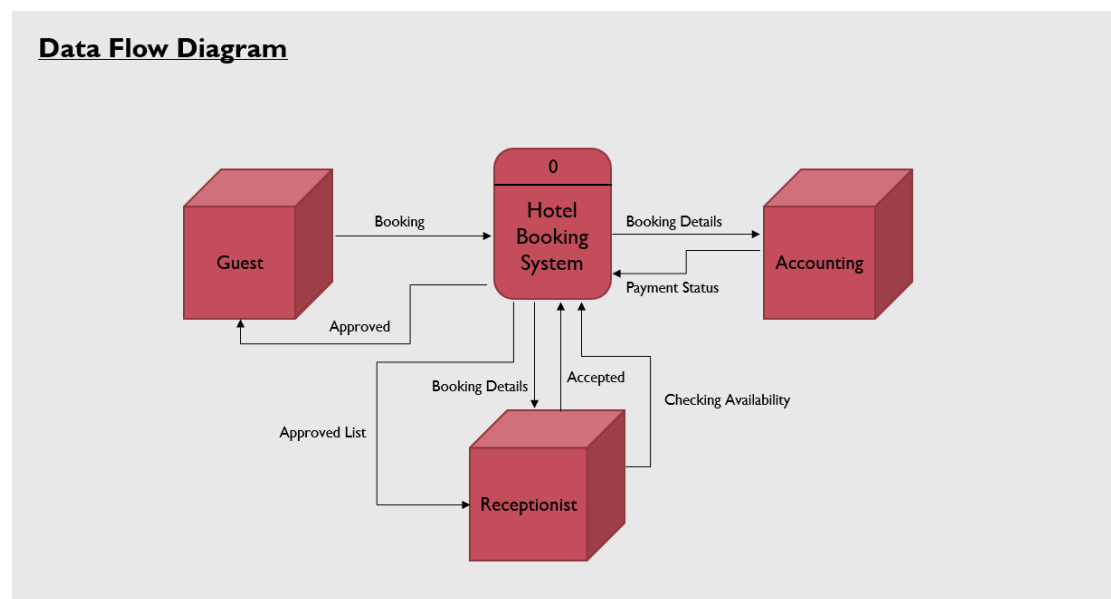
The system I designed for this class project is a hotel reservation system. I work for a company that sends technicians across the southern part of this country to work on servicing and repairing cranes. Part of my job is to make sure our technicians have a place to stay when they travel. As you might can imagine, I conduct lots of business with several different hotels across the country. Since I was already knowledgeable about typical procedures from the customer standpoint, I figured it would be a good system to try and design.

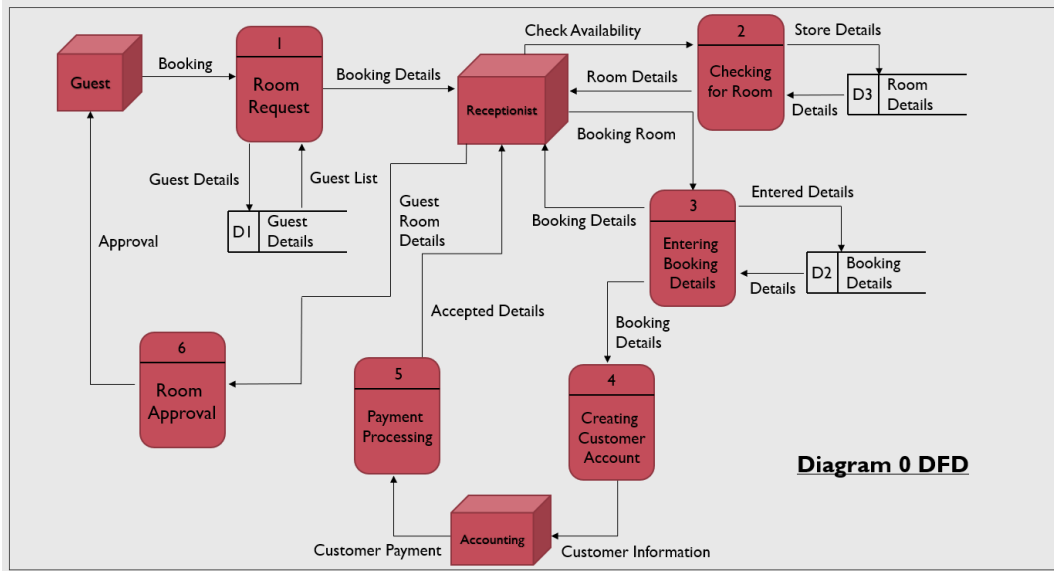
To start, my central process is of course the hotel booking system. This system shares data flow between three external entities (guest, receptionist, and accounting). Originally, I was going to have management as the external entity instead of accounting, but in my experience, management and accounting share some of the same duties so I went with what I knew more about. Essentially, my data flow diagram shows all the interactions between the process and my three external entities. I wanted to illustrate the steps taken by a guest who contacts a hotel to reserve a room. My project did NOT include online reservations. This was strictly person-to-person interactions.

My goal was to make this system as main steam as possible. As someone that must work with hotels on a daily basis, it is essential to be able to reserve rooms quickly and efficiently. I wanted to design a system which would enable me to do that. And since I'm the type of person calling from a business, it's much easier to handle the billing side of things by calling and speaking with reception at the hotel directly (since most hotels require you to give a credit card to make a reservation). I wouldn't say this system is improving existing ones, or that it's brand new. There are some hotels out there that work the same way I've designed this system. Then again, there are others out there that far over complicate things and I wind up spending 15 minutes on the phone with. My system is how I wish all the more complicated hotels would be like.

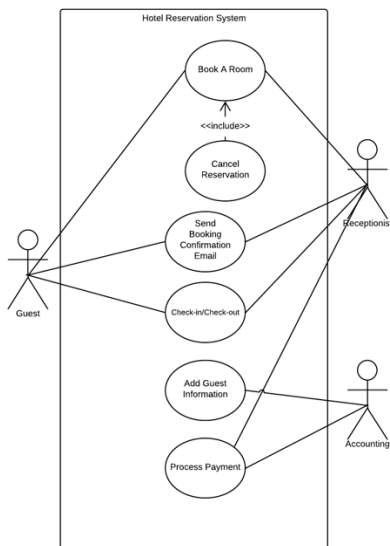
Imagine needing to call five different hotels to make extended reservations for several different technicians. Now, imagine explaining your needs to a receptionist only to be forwarded to a call center where people with language barriers proceed to ask you a multitude of questions about why you're calling so you must explain it all over again. And then once you finally share all the details and give them all of the information they require, you are then told they can't help you and that you'll need to call the hotel back and speak with them directly. From a business standpoint, that was a huge waste of time, right? It's also rather frustrating. This system I designed is meant to avoid that frustration and save you time in the long run.

System Requirements Model





As you can see, my process starts with the hotel booking system. My first external entity is the guest. The guest expresses interest in reserving a room with the hotel. The second external entity (receptionist) is presented with the booking details of the guest who then checks availability. The system sends a list matching the booking details and the receptionist chooses one to match what the guest asked for. Once the room is agreed upon, the details are sent to accounting for payment. Accounting then must create a guest profile (or account) in which payment information can be stored. After the payment is satisfied, payment status is sent back to the receptionist and the approval notification is sent to the guest.



USE CASE DIAGRAM

Above, you'll see my Use Case Diagram. I have six use cases: book a room, cancel reservation, send booking confirmation email, check-in/check-out, add guest information, and process payment. There are three actors: guest, receptionist, and accounting.

UC1: Book A Room

- Scope: Hotel Reservation System
- Level: User-goal Level Use Case
- Primary Actor: Guest
- Main Success Scenario:
 1. Guest expresses interest in booking a room.
 2. Guest provides identification for account setup.
 3. There is a room available.
 4. Guest provides credit card for reservation.
- Alternate Scenarios:

2a. Guest already has account.

1. Confirm guest's identity matches account on file.

3a. There is no room matching guest's needs available.

1. Inform guest of what rooms are available and ask if any of them would work instead.
2. Recommend partnered hotels that are nearby.

4a. Credit card is declined.

1. Inform guest their card was declined and ask for alternate payment method.

- Preconditions:
 - Guest walks into hotel or calls ahead of time.

- Rooms are available.
- Postconditions:
 - Guest receives confirmation of room reservation.
 - Guest's account is stored for future use.

UC2: Cancel A Reservation

- Scope: Hotel Reservation System
- Level: User-goal Level Use Case
- Primary Actor: Guest
- Main Success Scenario:
 1. Guest previously reserved a room.
 2. Guest cancels within appropriate time frame.
 3. Guest's credit card is refunded if charged.
- Alternate Scenarios:
 - 1a. There is no room reserved under guest's information.
 1. Ask guest to confirm their information for accuracy.
 2. Inform guest there is no record of their reservation.
 - 2a. Guest cancels after their cancelation window has gone by.
 1. Apologize and inform guest of hotel cancelation policy.
 - 2b. Guest was not informed of cancelation policy by whomever booked their room.
 1. Contact manager on duty and ask how they wish to proceed.
 1. Inform guest their card was declined and ask for alternate payment method.
- Preconditions:

- Guest reserved a room.
- Guest wishes to cancel room reservation.
- Postconditions:
 - Guest receives confirmation of room cancellation.
 - Guest's credit card is refunded.

UC3: Send Booking Confirmation

- Scope: Hotel Reservation System
- Level: User-goal Level Use Case
- Primary Actor: Receptionist
- Main Success Scenario:
 1. Guest successfully reserved a room.
 2. Guest's credit card was accepted.
 3. Guest provided an email for reservation confirmation.
- Alternate Scenarios:

3a. Guest did not provide an email address.

1. Print confirmation and hand to guest upon check-in.

- Preconditions:
 - Guest reserved a room.
 - Guest provided a credit card.
- Postconditions:
 - Guest is made aware of how much their credit card will be charged for.

UC4: Check-in/Check-out

- Scope: Hotel Reservation System
- Level: User-goal Level Use Case
- Primary Actor: Guest
- Main Success Scenario:
 1. Guest arrives to hotel.
 2. Guest has a room reserved.
 3. Guest confirms identity and receives room key.
 4. Guest returns room key upon finishing their stay.
- Alternate Scenarios:

2a. There is no room reserved matching guest's identity.

1. Ask for any other name the room could be under.
2. Ask guest when they made the reservation and who they spoke to.
3. Apologize for inconvenience and try to find them a new room.

3a. Guest cannot confirm their identity.

1. Inform guest of hotel policies and explain why identification verification is necessary.

4a. Guest doesn't return room key upon check-out.

1. Remind guest of the fee associated with failing to return room keys.
2. Charge guest's credit card for said fee.

- Preconditions:
 - Guest arrives at hotel.
 - Guest has room reserved.
- Postconditions:

- Guest is sent receipt for room charges.

UC5: Add Guest Information

- Scope: Hotel Reservation System
- Level: User-goal Level Use Case
- Primary Actor: Accounting
- Main Success Scenario:
 1. Guest provided their information for booking a room.
 2. Receptionist entered correct information from guest.

- Alternate Scenarios:

1a. Guest gave inaccurate information.

1. Notify reception of error and have them contact the guest to get correct information.

2a. Receptionist entered wrong information.

1. Notify reception of error and ask for correct information.

- Preconditions:
 - Guest expressed interest in booking a room.
 - Guest provided their information for account setup.
- Postconditions:
 - Guest's information is saved in the hotel's system.

UC6: Process Payment

- Scope: Hotel Reservation System
- Level: User-goal Level Use Case
- Primary Actor: Accounting

- Main Success Scenario:

1. Guest books a room.
2. Guest provides credit card information.

- Alternate Scenarios:

2a. Guest provides inaccurate credit information if booking via telephone.

1. Contact guest and ask for them to repeat their card information for verification.
2. Ask for different method of payment.

2b. Receptionist enters incorrect credit card information.

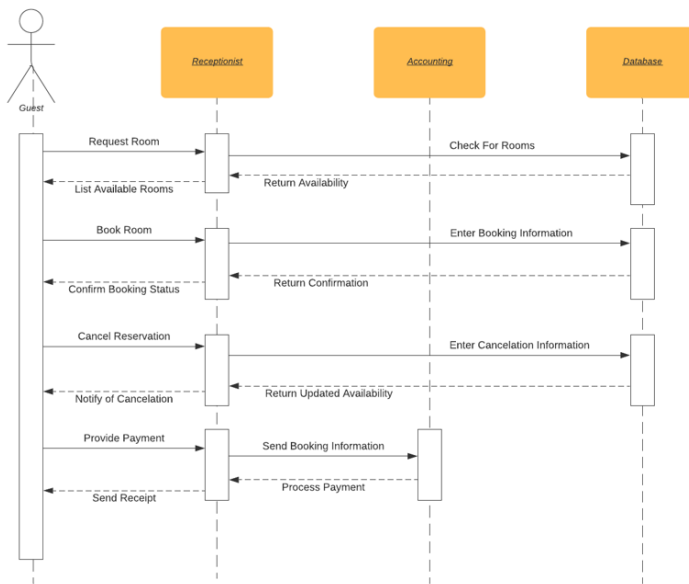
1. Inform receptionist of error and have them retrieve correct card information.

- Preconditions:

- Guest wishes to book a room.
- Guest provides hotel with their credit card to hold the room.

- Postconditions:

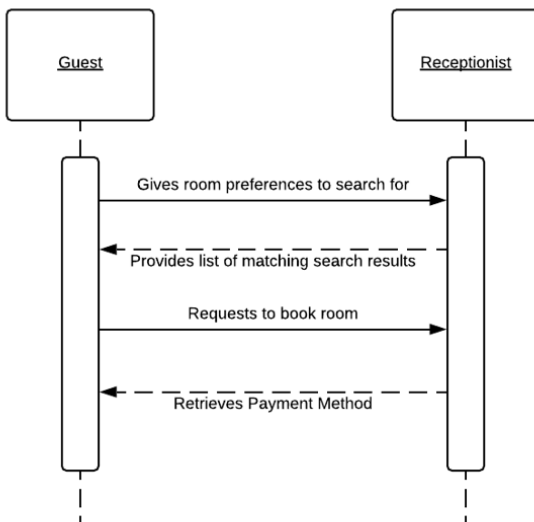
- Guest receives receipt of credit payment.



SYSTEM SEQUENCE DIAGRAM

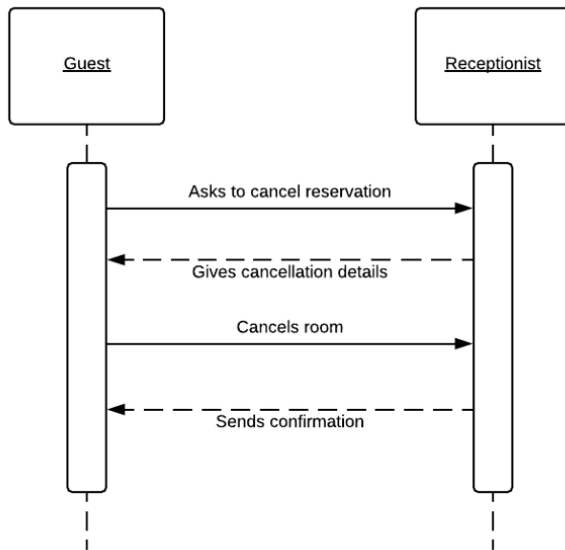
The above is my system sequence diagram. As you can see, I used some of the same text from my use cases to stay on track. We have the three actors interacting with the main hotel data base. This essentially shows a more in-depth look at how these actors are interacting from the Use Case Diagram and how they play into the hotel reservation system.

Below, you'll find each sequence diagram listed individually.



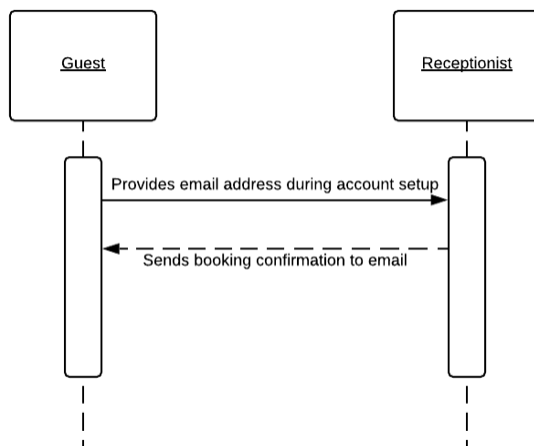
SD I – BOOK A ROOM

Here, you see how the guest and receptionist interact when trying to book a room. The guest expresses interest in booking a room, the receptionist collects the details and provides possibilities. The guest chooses a room, and the receptionist then retrieves payment method.



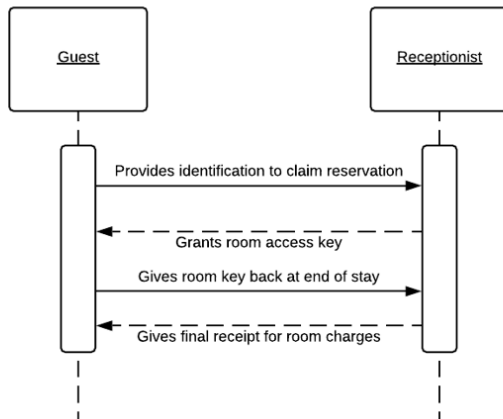
SD 2 – CANCEL RESERVATION

This SD is for the exception to booking a room. If the guest wants to cancel their reservation, they contact reception and ask to cancel. Reception gives the guest the details to cancel and if the guest still wishes to, they can and then they are sent a confirmation of their cancellation.



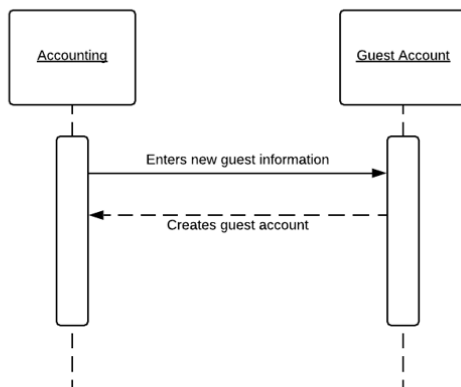
SD 3 – SEND BOOKING CONFIRMATION EMAIL

This SD is fairly straightforward. It simply is the interaction between the guest providing their email address to the receptionist so that they may receive and confirmation email upon their successful booking.



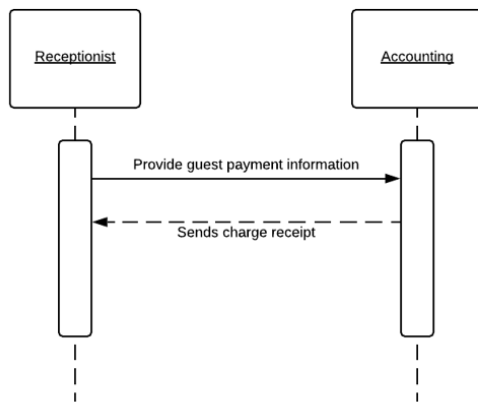
SD 4 – CHECK-IN/CHECK-OUT

This SD outlines the process of checking in and out of a room. The guest must provide identification upon checking in for the first time in order to get their key card. They then must return the key card upon checking out in order to get their final receipt.



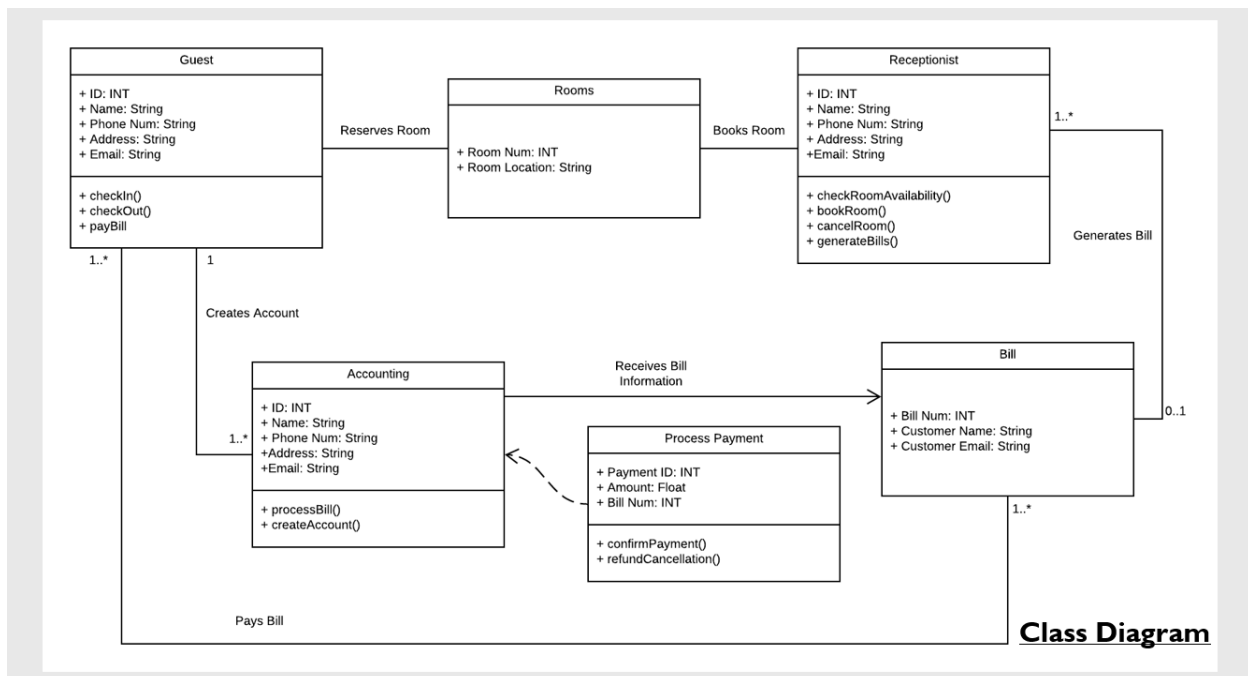
SD 5 – ADD GUEST INFORMATION

This is another pretty simple SD. Accounting must collect guest information in order to then create their account.



SD 6 – PROCESS PAYMENT

Accounting gets payment information from reception and then sends back payment status.



I'll be honest, the class diagram was the hardest for me to do. What was discussed in class didn't make much sense to me, nor did the lecture notes. I sought help from the internet and it didn't do much for me either. I saw so many different ways to do the class diagram. I'm sure I've done it wrong. But, I did what I could. I'm fairly sure my process payment was dependent on accounting. Because without accounting, you couldn't process a payment. And

Accounting was associated with the bill because they have to use it to process payments. I feel like the others are all inherently associated with one another. The guest is associated with rooms because they are booking it, and the rooms are associated with the receptionist because they have to reserve it. Reception is associated with the bill because they originally generate it for accounting. That's how I saw this. I know I didn't do the arrows correctly. Sorry about that.

Lessons Learned

What would I do differently? I would hopefully actually get a partner to help me with this. Someone to bounce ideas off of. If I had a partner, I probably would have included the online reservation portion of what would make a full hotel reservation system. But, since it was just me, I had to keep it simple.

Team Description

I completed this project all by my lonesome. No one else to blame for mistakes but me!

