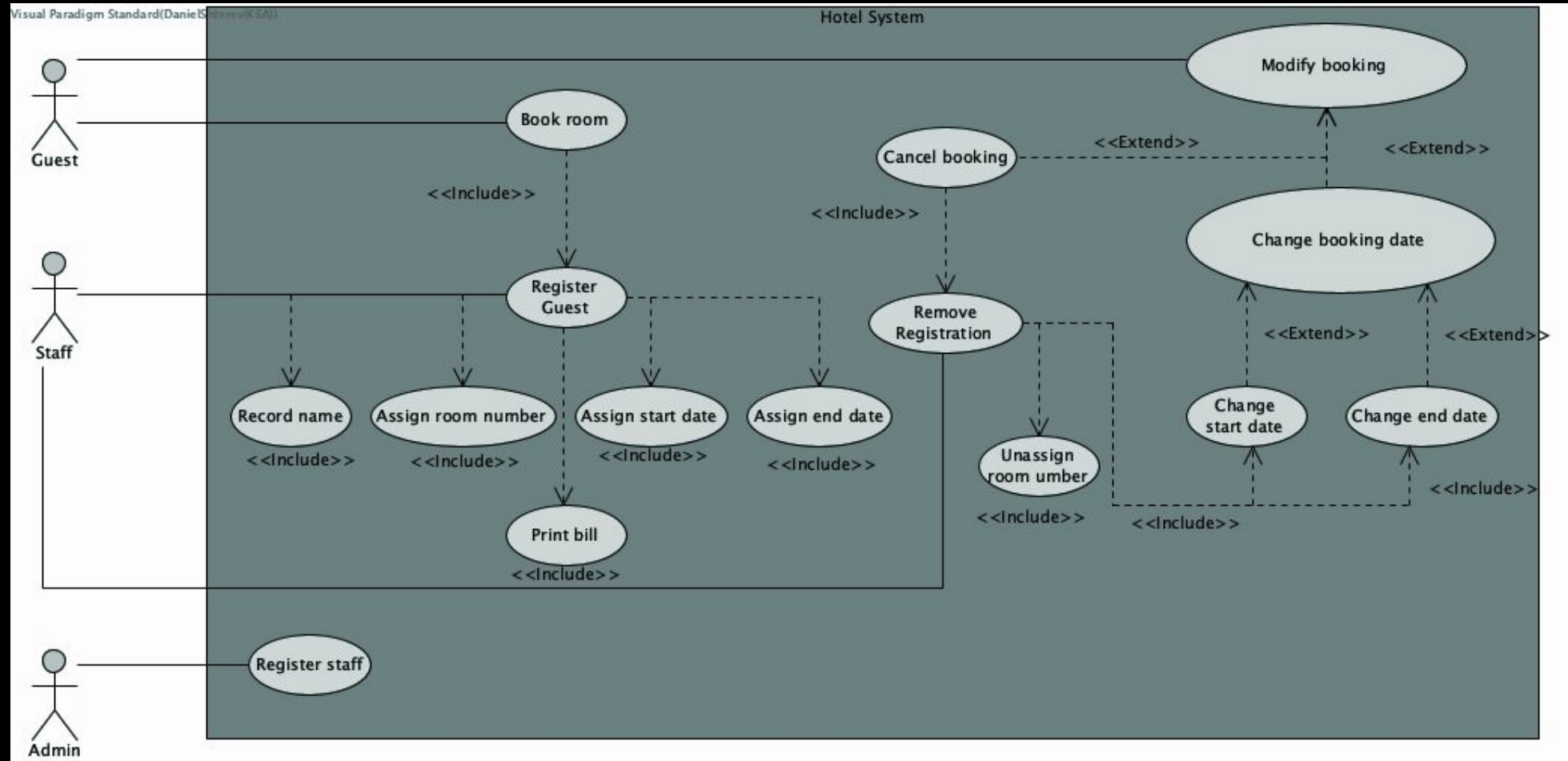




Kevin Williams  
Daniel Shterev  
Victoria Velichkova  
Ventsislav Iliev

# KAIZEN'S HOTEL - MODULE 3

# USE CASE



# USE CASE FULL DRESS

USE CASE: Hotel application  
Primary Actor: Staff

Stakeholders and Interests:

- Guest: Wants to book a room in the hotel
- Staff: Registers guests
- Admin: Registers staff within the program

Preconditions:

- Staff logs into program

Success Guarantee (post-conditions): Staff successfully logs into application and hotel database is loaded.

Main Success Scenario (basic flow):

1. Staff registers and logs into application.
2. Hotel application confirms login.
3. Staff registers guest to a room.
4. Hotel database is updated and saved.
5. Guest is successfully registered to a room of their choice.
6. Database is saved.

Extensions (or Alternative Flows):

\*a. Staff fails to login.

1. User inputs "help" command in Hotel application
2. Staff attempts to login again.
- 2a. Staff login is successful.

Special Requirements:

- Hotel application has rooms hard-coded to database
- Staff members are registered

Technology and Data Variation List:

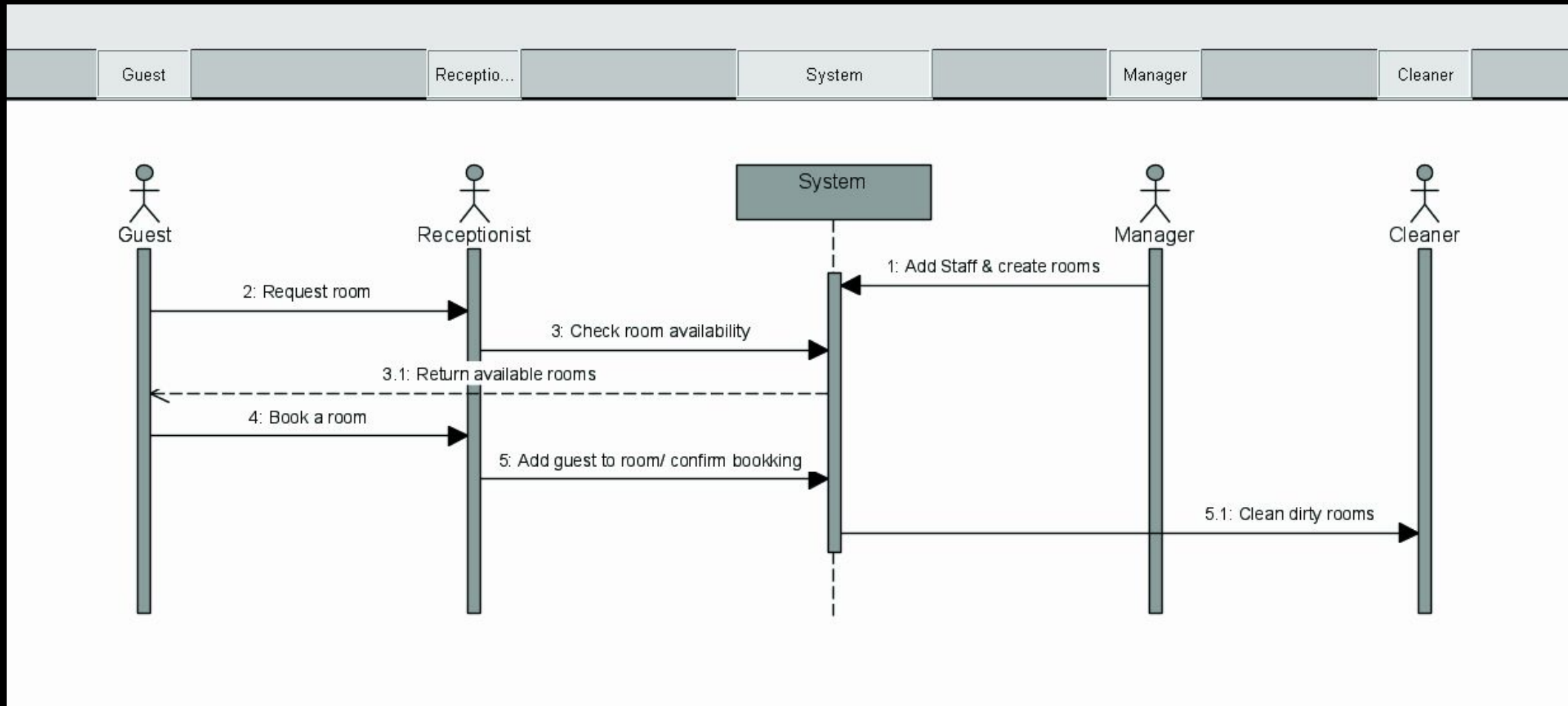
Frequency of Occurrences:

- Nearly continuous

Open Issues:

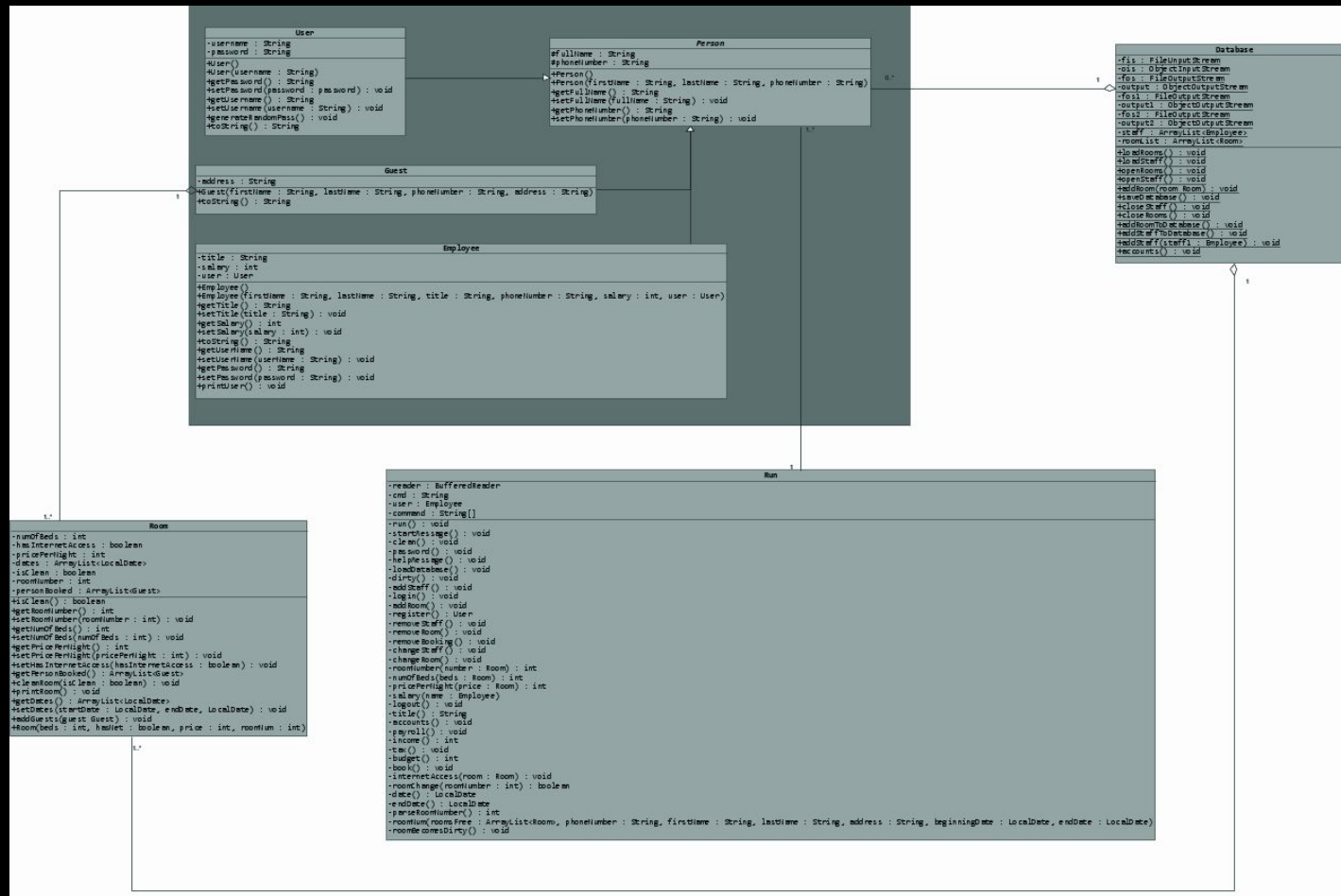
- Application maintenance
- Application security

# SYSTEM SEQUENCE DIAGRAM

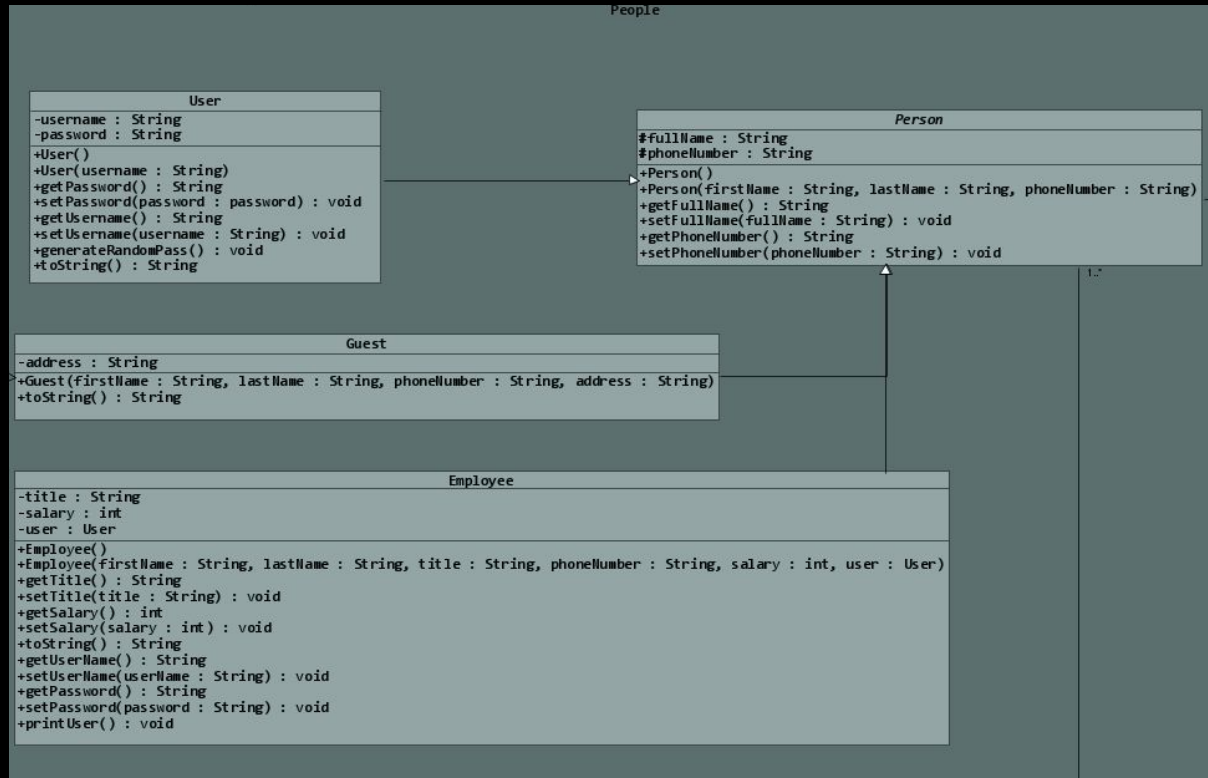




# CLASS DIAGRAM



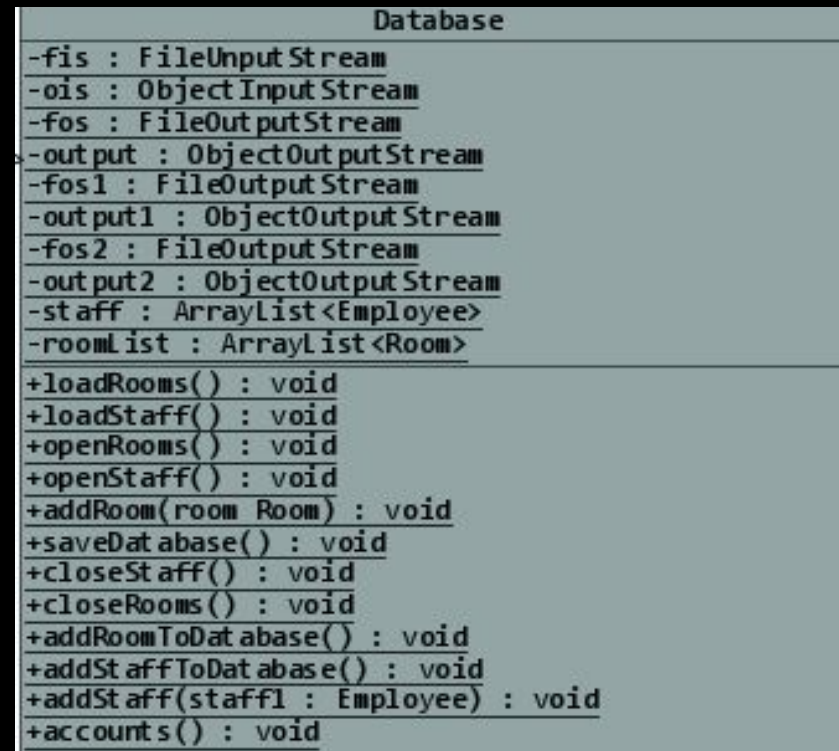
# CLASS DIAGRAM “PEOPLE”



# CLASS DIAGRAM “ROOM”

```
Room
- numofBeds : int
- hasInternetAccess : boolean
- pricePerNight : int
- dates : ArrayList<LocalDate>
- isClean : boolean
- roomNumber : int
- personBooked : ArrayList<Guest>
+ isClean() : boolean
+ getRoomNumber() : int
+ setRoomNumber(roomNumber : int) : void
+ getNumofBeds() : int
+ setNumofBeds(numofBeds : int) : void
+ getPricePerNight() : int
+ setPricePerNight(pricePerNight : int) : void
+ setHasInternetAccess(hasInternetAccess : boolean) : void
+ getPersonBooked() : ArrayList<Guest>
+ cleanRoom(isClean : boolean) : void
+ printRoom() : void
+ getDates() : ArrayList<LocalDate>
+ setDates(startDate : LocalDate, endDate, LocalDate) : void
+ addGuests(guest Guest) : void
+ Room(beds : int, hasNet : boolean, price : int, roomNum : int)
```

# CLASS DIAGRAM “DATABASE”

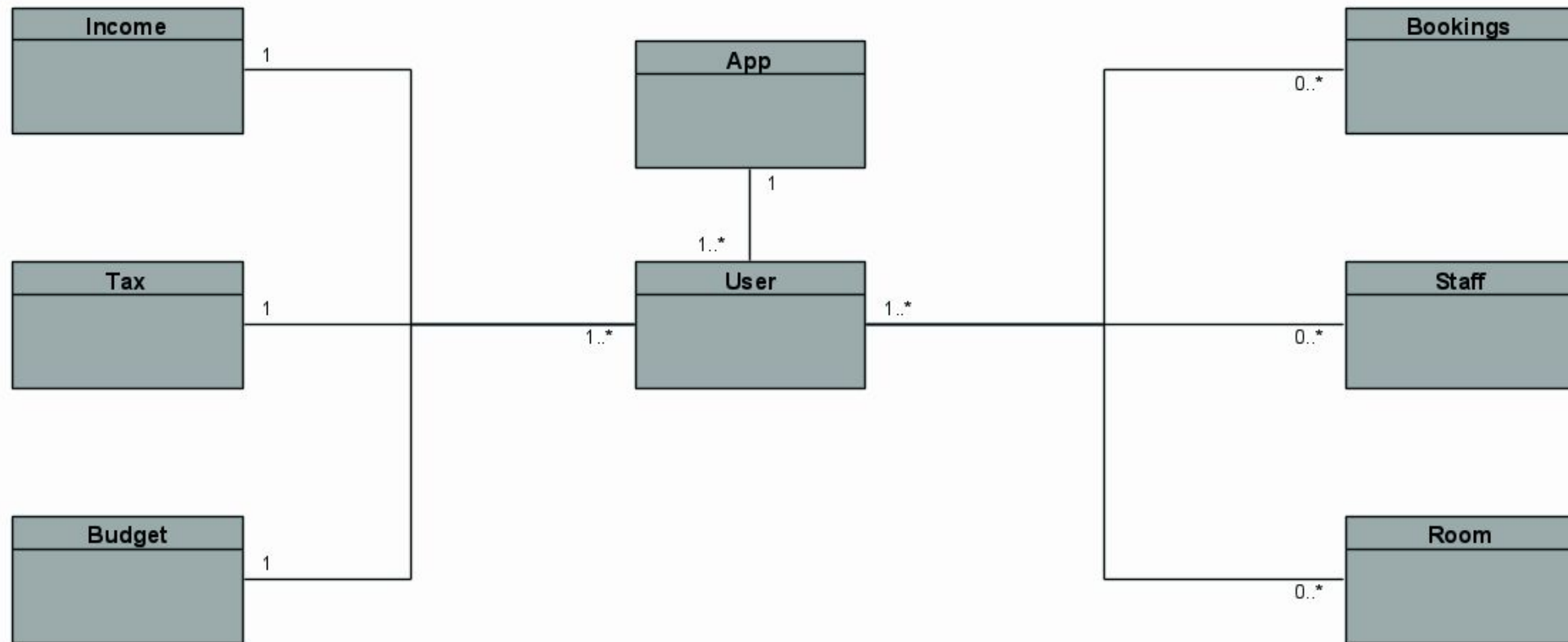




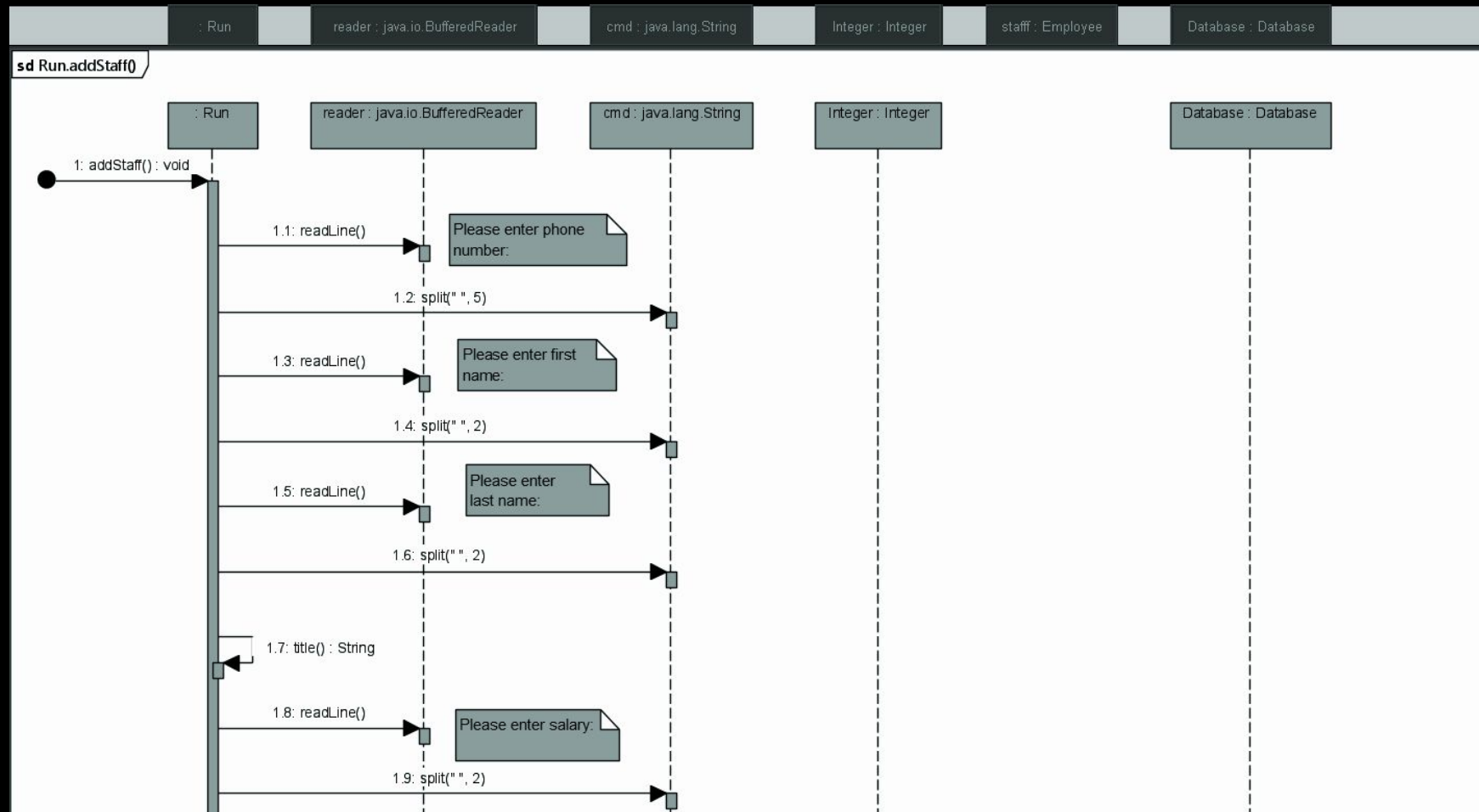
# CLASS DIAGRAM “RUN”

```
Run
- reader : BufferedReader
- cmd : String
- user : Employee
- command : String[]
- run() : void
- startMessage() : void
- clean() : void
- password() : void
- helpMessage() : void
- loadDatabase() : void
- dirty() : void
- addStaff() : void
- login() : void
- addRoom() : void
- register() : User
- removeStaff() : void
- removeRoom() : void
- removeBooking() : void
- changeStaff() : void
- changeRoom() : void
- roomNumber(number : Room) : int
- numofBeds(beds : Room) : int
- pricePerNight(price : Room) : int
- salary(name : Employee)
- logout() : void
- title() : String
- accounts() : void
- payroll() : void
- income() : int
- tax() : void
- budget() : int
- book() : void
- internetAccess(room : Room) : void
- roomChange(roomNumber : int) : boolean
- date() : LocalDate
- endDate() : LocalDate
- parseRoomNumber() : int
- roomNum(roomFree : ArrayList<Room>, phoneNumber : String, firstName : String, lastName : String, address : String, beginningDate : LocalDate, endDate : LocalDate)
- roomBeComesDirty() : void
```

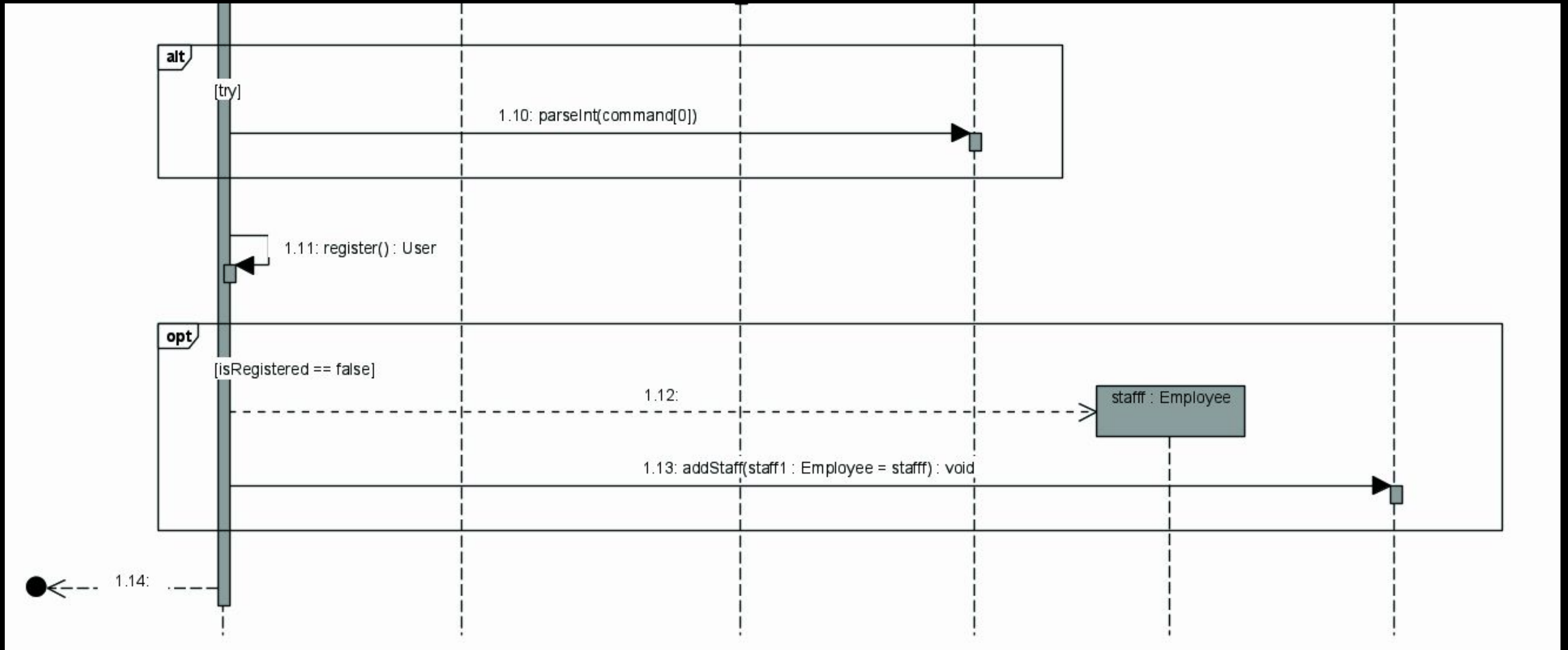
# DOMAIN MODEL



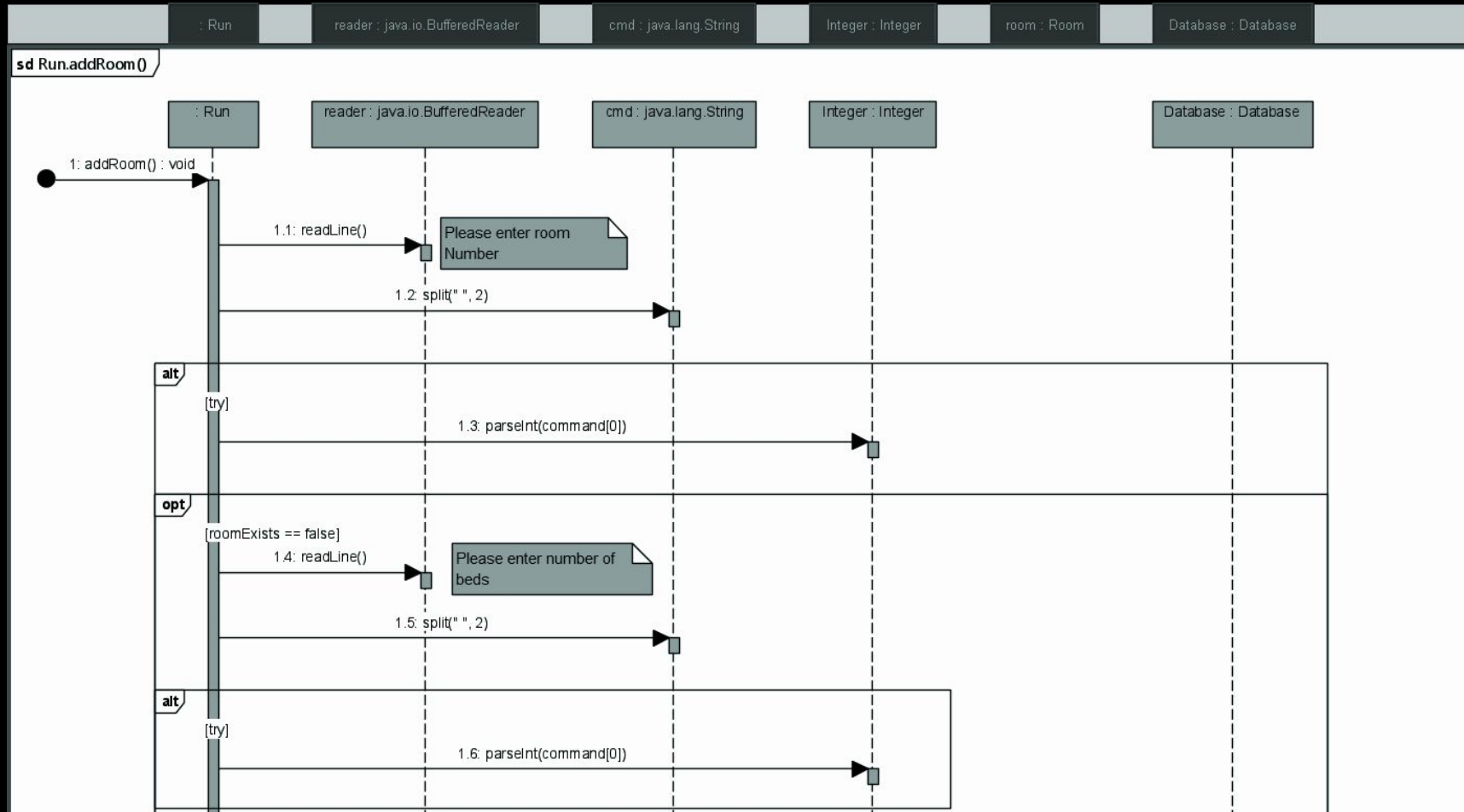
# SEQUENCE DIAGRAM “addStaff”



# “addStaff” Continued

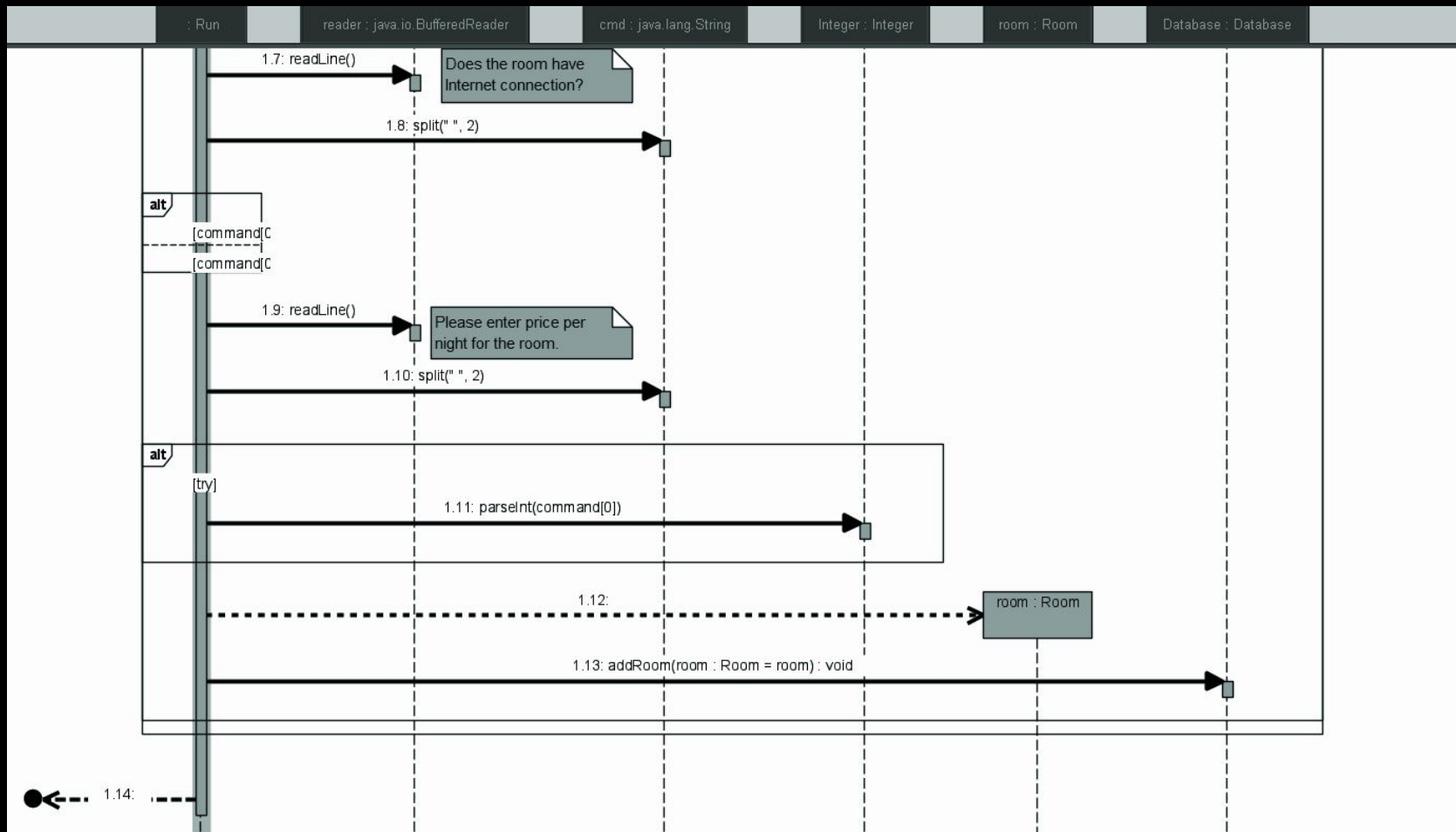


# SEQUENCE DIAGRAM “addRoom”

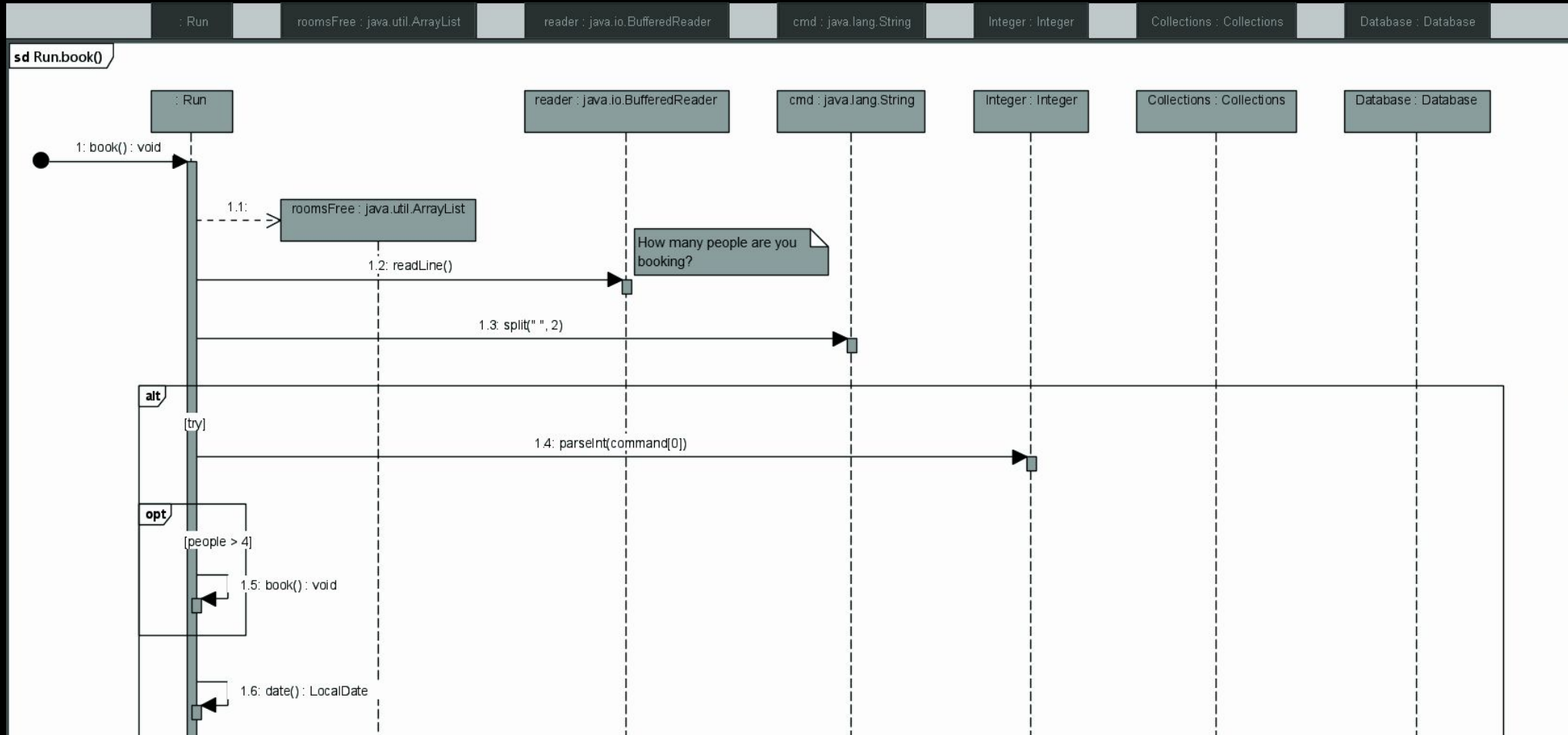




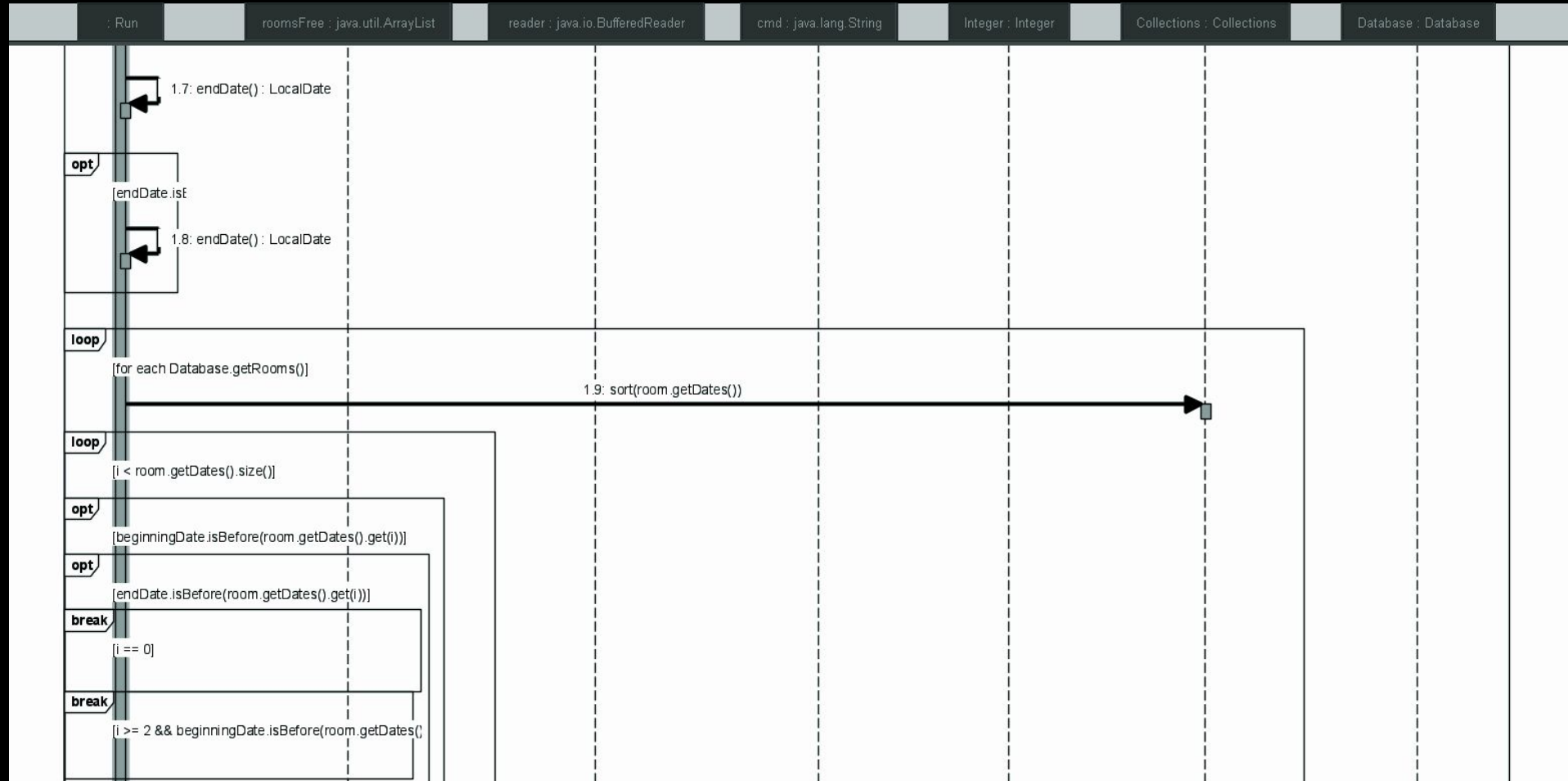
# “addRoom” Continued



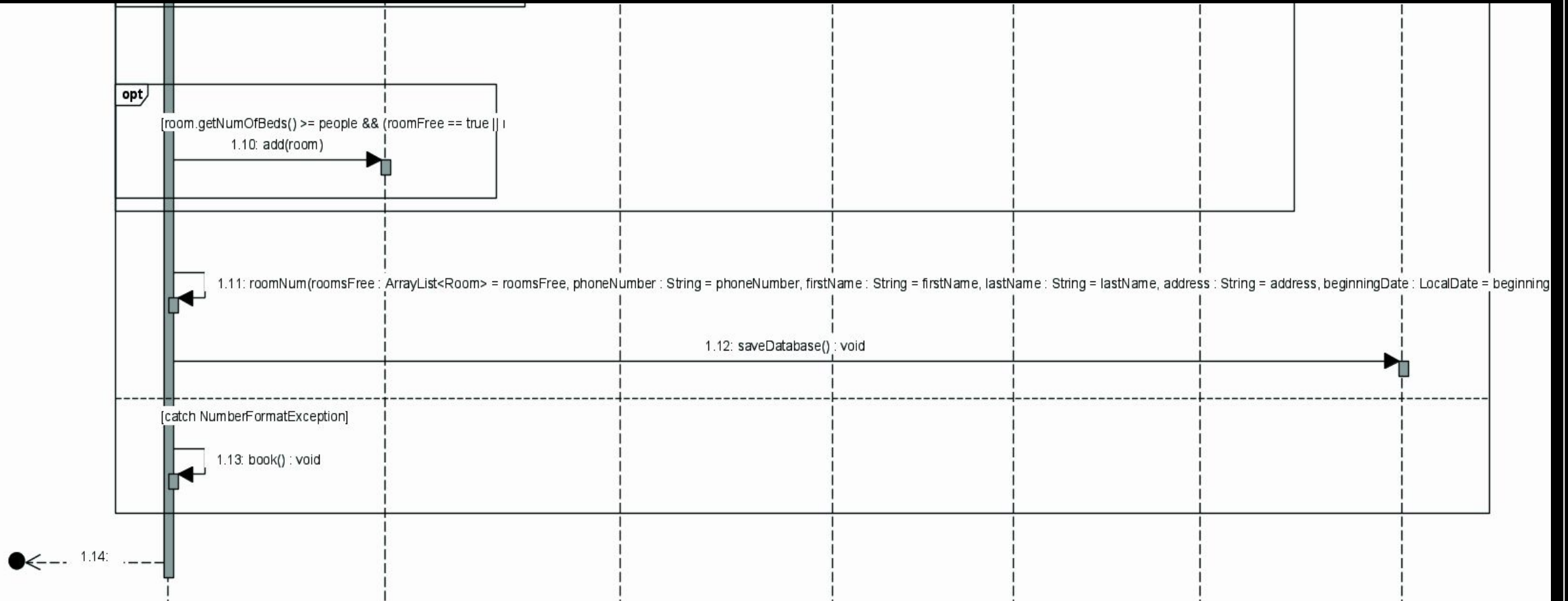
# SEQUENCE DIAGRAM “book”



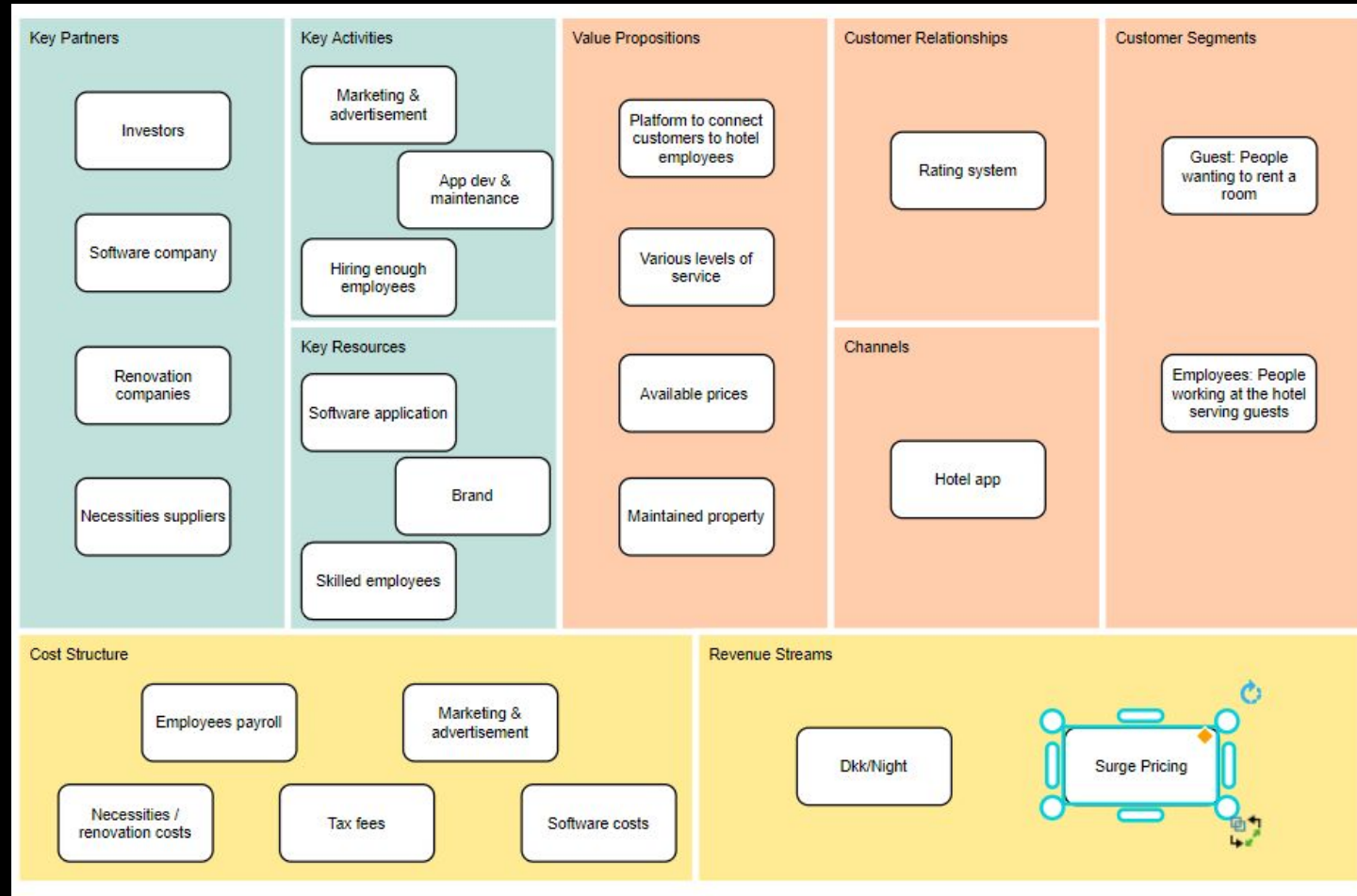
# “book” Continued



# “book” Continued

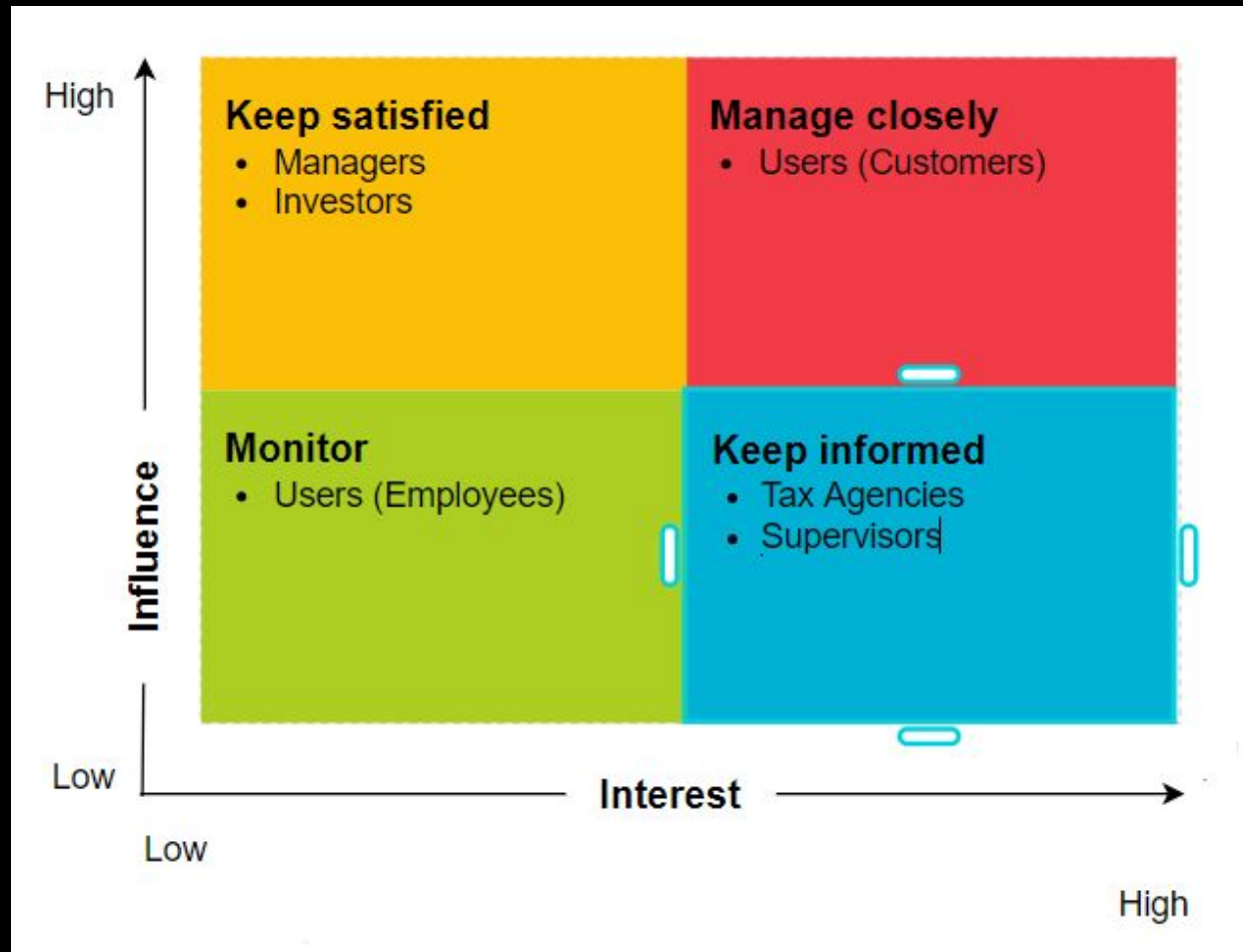


# Business Model Canvas

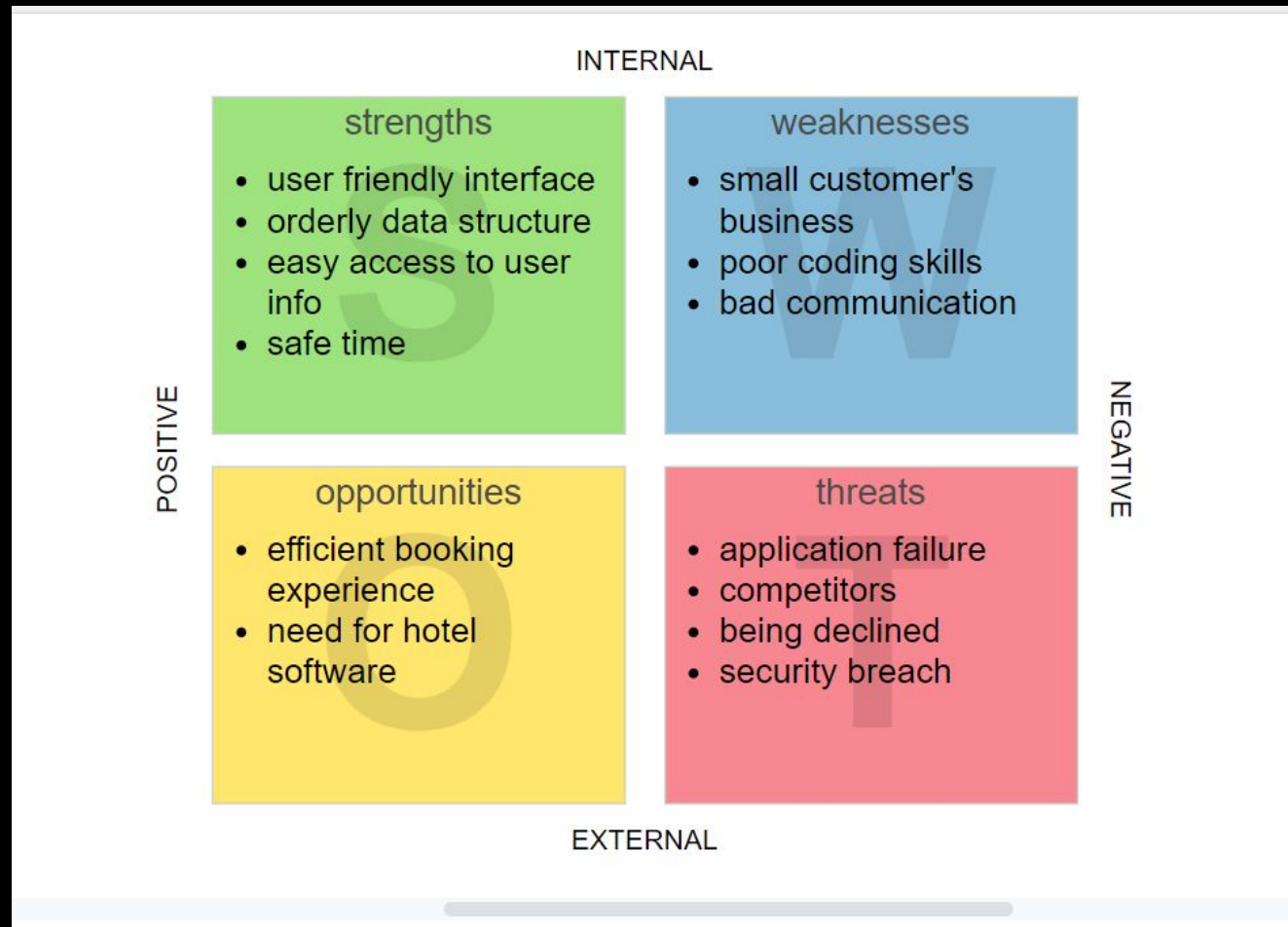




# STAKEHOLDER ANALYSIS



# SWOT ANALYSIS



# GANTT CHART

## Module 3

## Group Kaizen

Project Start:

Thu, 11/11/2021

Display Week:

1

[illegible]

# GANTT CONTINUED

[illegible]



TO THE CODE!