

PA2550: Seminar Series in Software Engineering

Group Project: Design Documentation

30-09-2024

Student 1	Name	Makarudze, Anna
	Email	anmk24@student.bth.se
	Swedish ID	860630T441
Student 2	Name	Mahapatra, Payel
	Email	pamh24@student.bth.se
	Swedish ID	930618T729
Student 3	Name	Kotapati, Jammithri
	Email	jako22@student.bth.se
	Swedish ID	20010610T223
Student 4	Name	Rokaya, Sanjeeb
	Email	saro24@student.bth.se
	Swedish ID	010702T339
Student 5	Name	Arakkal Sudesh, Meenakshi
	Email	meaa24@student.bth.se
	Swedish ID	951124R029

1. UML Interaction Diagrams

Use Case Name:	Make Room Reservation
Use Case ID:	UC01
Actor(s):	Guests, Manager, Front Desk Staff
Type:	Primary
Basic Flow:	Guest, Manager, and Front Desk Staff who want to make a reservation for a room will visit the Bed and Breakfast website. The logged in Guest, Manager and the Front Desk Staff can choose the desired check-in and check-out dates and the number of guests. The website will display the available rooms on the specified dates. The Guest, Manager and Front Desk Staff view individual rooms and click on the “Reserve” button against a desired room. The Guest, Manager and Front Desk Staff are redirected to a booking page where they enter the guest’s details and book a room. The Guest will receive the booking confirmation and a unique booking reference number through email.

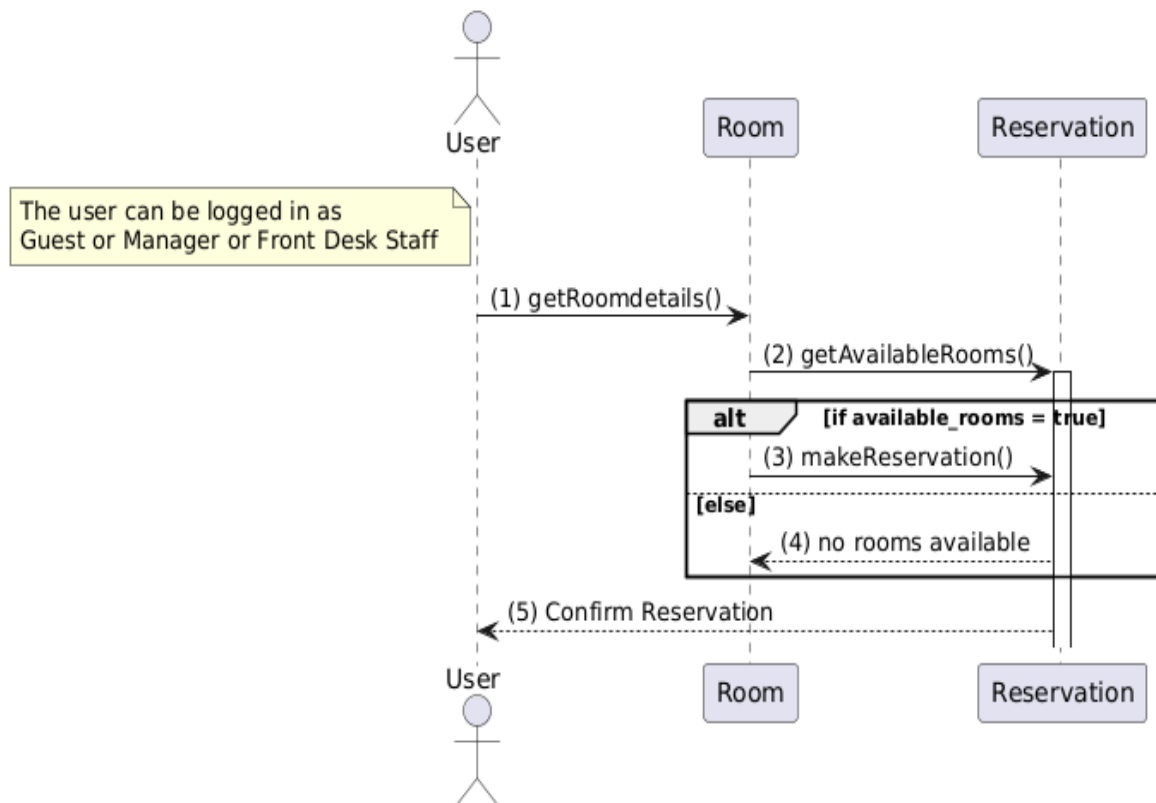


Figure 1: Reserve a Room Sequence Diagram

```

@startuml
autonumber "(#)"
actor User
note left of User
The user can be logged in as
Guest or Manager or Front Desk Staff
end note
User -> Room: getRoomdetails()
Room -> Reservation: getAvailableRooms()
activate Reservation
alt if available_rooms = true
Room -> Reservation: makeReservation()
else else
Reservation --> Room: no rooms available
end
Reservation --> User: Confirm Reservation
@enduml
  
```

Use Case Name	Viewing reports
Use Case ID	UC02
Actor(s)	Manager
Type	Primary
Basic Flow	The Manager selects the period for which he wants to view the report. The System processes the request and returns a summary of the BnB reservations based on the selected period.

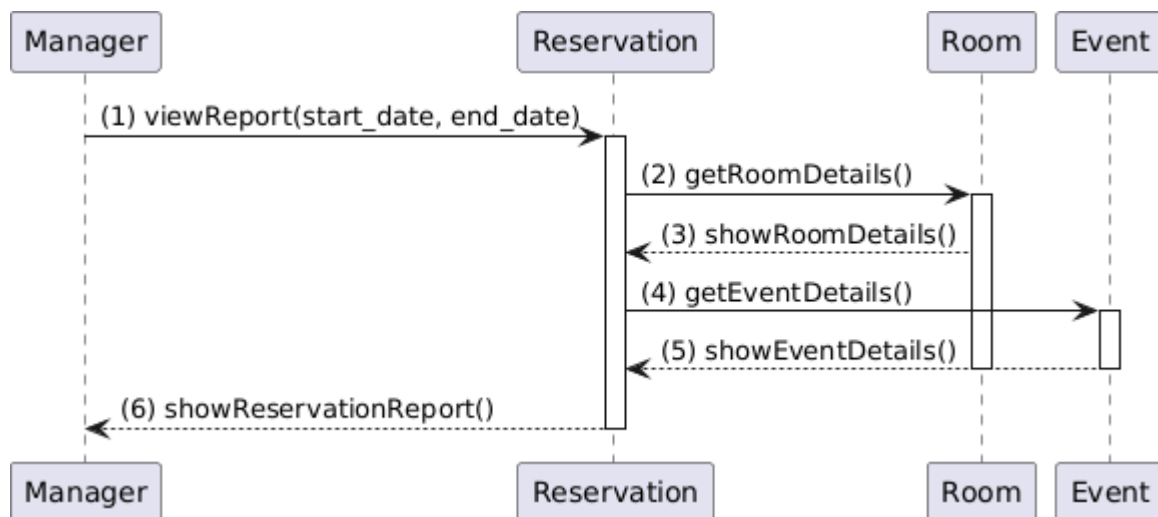


Figure 2: View Reports Sequence Diagram

@startuml

participant Manager as Foo1

participant Reservation as Foo2

participant Room as Foo3

participant Event as Foo4

autonumber 1 "(#)"

Foo1 -> Foo2 : viewReport(start_date, end_date)

activate Foo2

Foo2 -> Foo3 : getRoomDetails()

activate Foo3

Foo3 --> Foo2 : showRoomDetails()

Foo2 -> Foo4 : getEventDetails()

activate Foo4

Foo4 --> Foo2 : showEventDetails()

deactivate Foo3

deactivate Foo4

Foo2 --> Foo1 : showReservationReport()

deactivate Foo2

@enduml

2. UML Class Diagrams

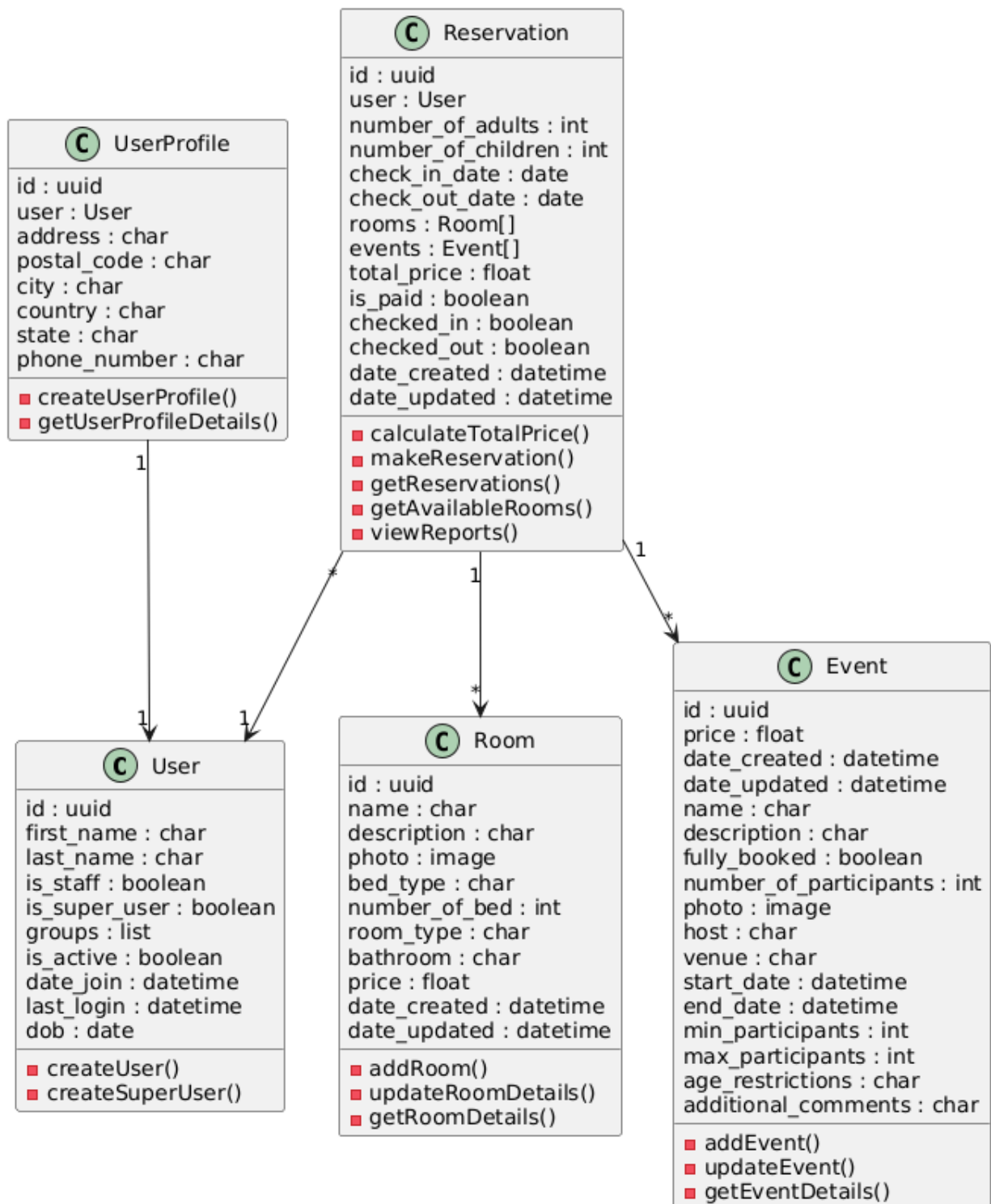


Figure 3: BnB System Class Diagram

@startuml

```
class User {  
    id : uuid  
    first_name : char  
    last_name : char  
    is_staff : boolean  
    is_super_user : boolean  
    groups : list  
    is_active : boolean  
    date_join : datetime  
    last_login : datetime  
    dob : date
```

```
-createUser()  
-createSuperUser()  
}
```

```
class UserProfile {  
    id : uuid  
    user : User  
    address : char  
    postal_code : char  
    city : char  
    country : char  
    state : char  
    phone_number : char
```

```
-createUserProfile()  
-getUserProfileDetails()  
}
```

```
class Reservation {  
    id : uuid  
    user : User  
    number_of_adults : int  
    number_of_children : int  
    check_in_date : date  
    check_out_date : date
```

```
rooms : Room[]
events : Event[]
total_price : float
is_paid : boolean
checked_in : boolean
checked_out : boolean
date_created : datetime
date_updated : datetime
```

```
-calculateTotalPrice()
-makeReservation()
-getReservations()
-getAvailableRooms()
-viewReports()
```

```
}
```

```
class Room {
  id : uuid
  name : char
  description : char
  photo : image
  bed_type : char
  number_of_bed : int
  room_type : char
  bathroom : char
  price : float
  date_created : datetime
  date_updated : datetime
```

```
-addRoom()
-updateRoomDetails()
-getRoomDetails()
```

```
}
```

```
class Event{
  id : uuid
  price : float
```


date_created : datetime
date_updated : datetime
name : char
description : char
fully_booked : boolean
number_of_participants : int
photo : image
host : char
venue : char
start_date : datetime
end_date : datetime
min_participants : int
max_participants : int
age_restrictions : char
additional_comments : char

-addEvent()
-updateEvent()
-getEventDetails()
}

UserProfile "1"--> "1" User
Reservation "*"--> "1" User
Reservation "1" --> "*" Room
Reservation "1" --> "*" Event
@enduml