Assignment 2 – Static Model
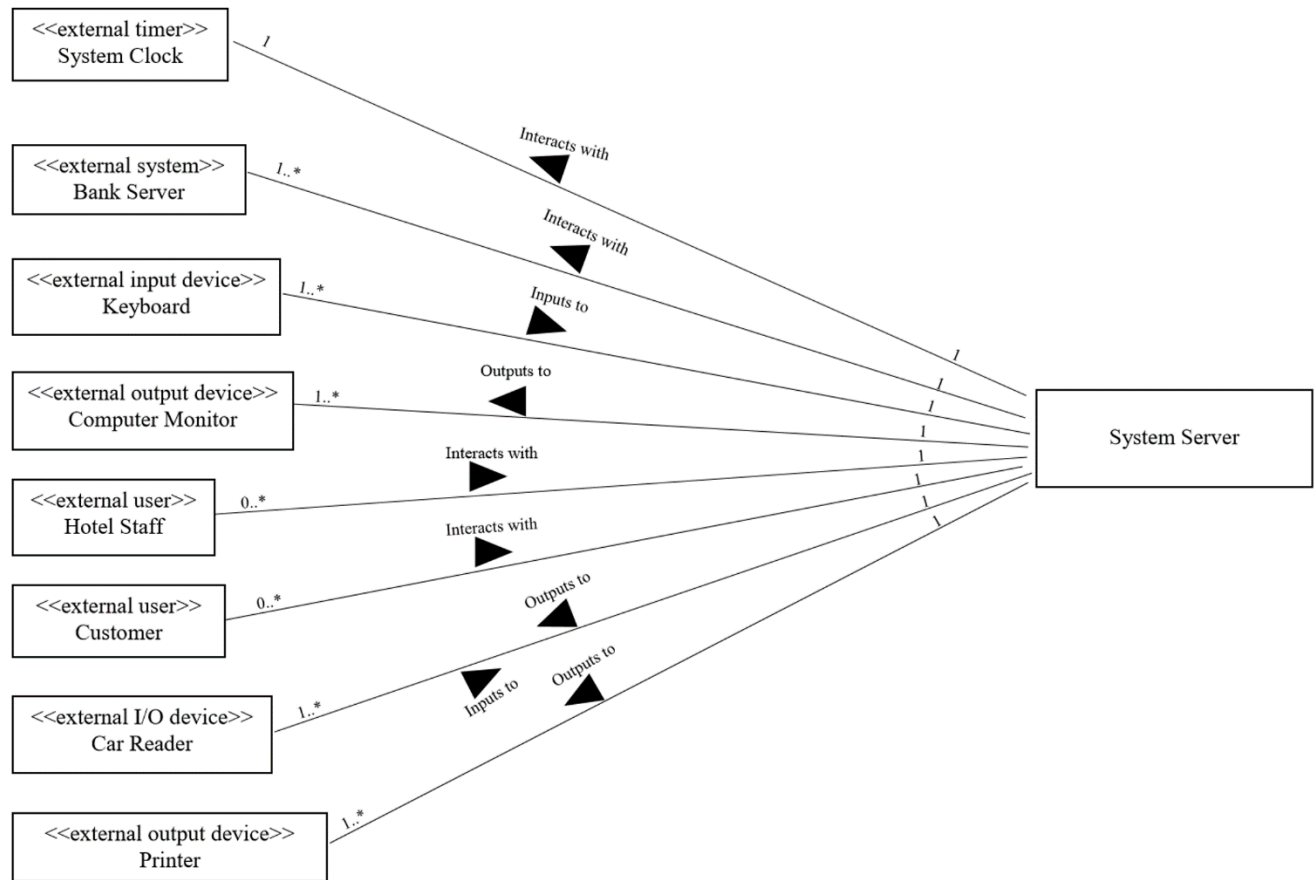
Courtney Olson, Marcus Domingo, Sam Kim

2. Software System Context Class Model
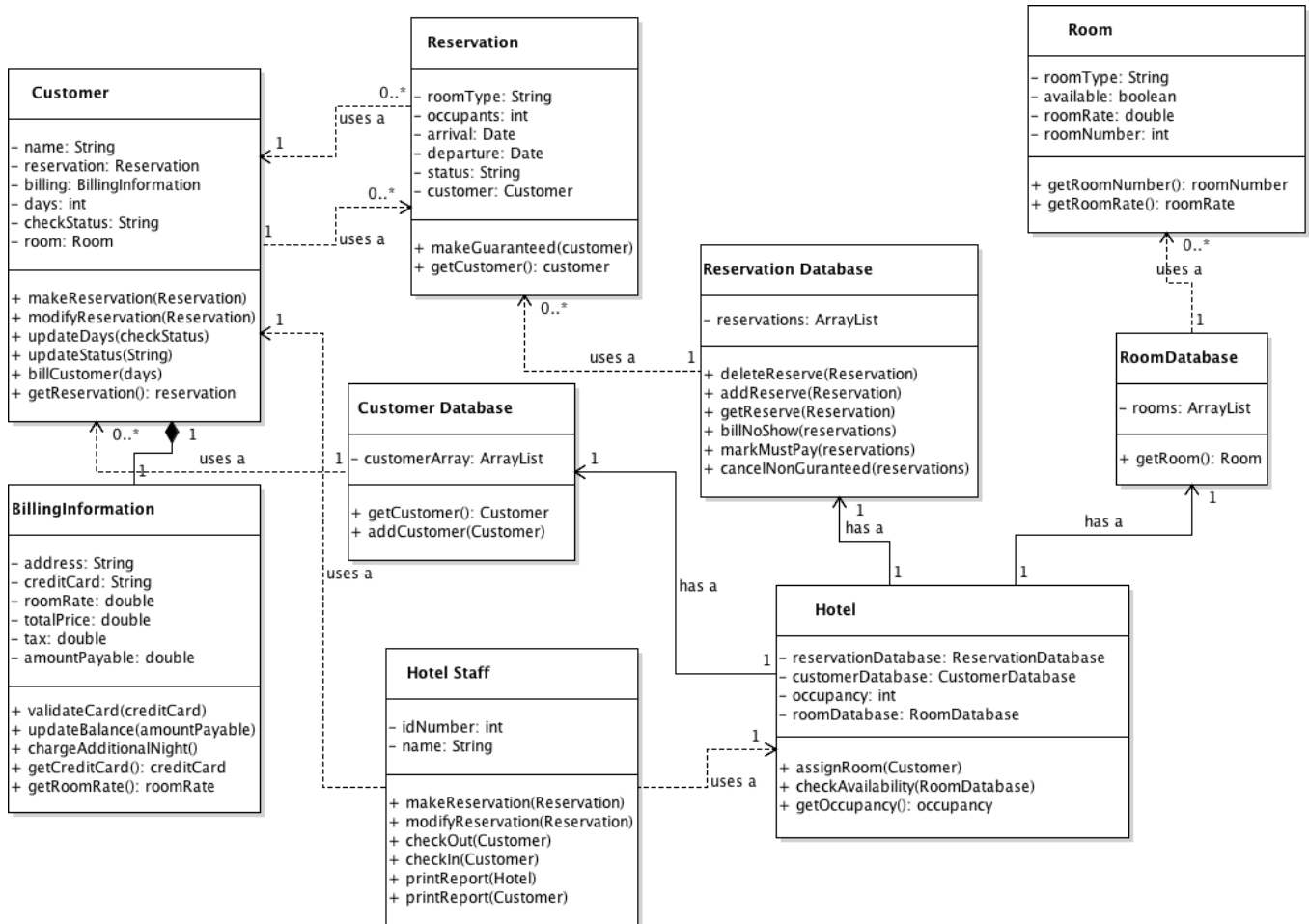
3. Static Model for Entity Classes

4. Purpose of Entity Classes

Software System Context Class Model

<<external timer>>
System Clock

<<external system>>
Bank Server

<<external input device>>
Keyboard

<<external output device>>
Computer Monitor

<<external user>>
Hotel Staff

<<external user>>
Customer

<<external I/O device>>
Car Reader

<<external output device>>
Printer

System Server

1

1..*

1..*

1..*

0..*

0..*

1..*

1..*

Interacts with

Interacts with

Inputs to

Outputs to

Interacts with

Interacts with

Inputs to

Outputs to

Outputs to

1

1

1

1

1

1

1

Static Model for Entity Classes

Reservation

- roomType: String
- occupants: int
- arrival: Date
- departure: Date
- status: String
- customer: Customer

+ makeGuaranteed(customer)
+ getCustomer(): customer

Room

- roomType: String
- available: boolean
- roomRate: double
- roomNumber: int

+ getRoomNumber(): roomNumber
+ getRoomRate(): roomRate

Customer

- name: String
- reservation: Reservation
- billing: BillingInformation
- days: int
- checkStatus: String
- room: Room

+ makeReservation(Reservation)
+ modifyReservation(Reservation)
+ updateDays(checkStatus)
+ updateStatus(String)
+ billCustomer(days)
+ getReservation(): reservation

Reservation Database

- reservations: ArrayList

+ deleteReserve(Reservation)
+ addReserve(Reservation)
+ getReserve(Reservation)
+ billNoShow(reservations)
+ markMustPay(reservations)
+ cancelNonGuranteed(reservations)

RoomDatabase

- rooms: ArrayList

+ getRoom(): Room

Customer Database

- customerArray: ArrayList

+ getCustomer(): Customer
+ addCustomer(Customer)

BillingInformation

- address: String
- creditCard: String
- roomRate: double
- totalPrice: double
- tax: double
- amountPayable: double

+ validateCard(creditCard)
+ updateBalance(amountPayable)
+ chargeAdditionalNight()
+ getCreditCard(): creditCard
+ getRoomRate(): roomRate

Hotel Staff

- idNumber: int
- name: String

+ makeReservation(Reservation)
+ modifyReservation(Reservation)
+ checkOut(Customer)
+ checkIn(Customer)
+ printReport(Hotel)
+ printReport(Customer)

Hotel

- reservationDatabase: ReservationDatabase
- customerDatabase: CustomerDatabase
- occupancy: int
- roomDatabase: RoomDatabase

+ assignRoom(Customer)
+ checkAvailability(RoomDatabase)
+ getOccupancy(): occupancy

uses a

has a

3

Purpose of Entity Classes

Customer: The Customer Class will store all information required for a given Customer (including an instance of the BillingInformation object and a Reservation object) and provide methods for modification of an individual Reservation, the Customer's fields, and billing the Customer through the use of BillingInformation.

BillingInformation: The BillingInformation Class will store all billing information required for a given Customer and provide methods for validating credit cards, updating balance, and adding additional charges to a Customer's balance.

CustomerDatabase: The CustomerDatabase will store all instances of Customer for the Hotel class and provide methods to add and get a given Customer.

Reservation: The Reservation Class will store all information required for a given Reservation (including an instance of the Customer object) and provide methods for Reservation status change and retrieving the corresponding Customer.

ReservationDatabase: The ReservationDatabase will store all instances of Reservation for the Hotel class and provide methods to modify a given Reservation and process billing for no-show guaranteed Reservations.

Room: The Room Class will store all information required for a given Room and provide methods for Room information retrieval.

RoomDatabase: The RoomDatabase will store all instances of Room for the Hotel class and provide a method to get a given Room's information.

HotelStaff: The HotelStaff Class will store all information required for a given HotelStaff and provide concrete methods for the HRS such as check-in, check-out, print report, and reservation modification.

Hotel: The Hotel Class will store all information about the HRS (such as Rooms, Reservations, and Customers) and provide methods to assign rooms, keep track of occupancy, and check availability.