Linnéuniversitetet
Kalmar Växjö

Task 1: Requirements Specification Document

Software Engineering - Design

- 2DV603

Member: Chaoran Meng
Member: Meng Li
Email: cm222in@student.lnu.se
Email: ml223tu@student.lnu.se

Contents

Domain Analysis Document

1.1 Introduction and Background

“Linnaeus Hotel Management” is the system that manages reception activities at “Linnaeus Hotel”. The legacy system is too complicated and not easy to update, and it lacks some new features. So our customers want us to design a modern and more efficient system. Modern and more efficient systems are more convenient and cost-effective than legacy system. And can better serve our guests. The system will be used to enter reservations and check in and out of the hotel guests.

1.2 Customers' goals and problems

The system should identify the number and type of each room. When the hotel staff entered the necessary information, the system must be validated in 2 seconds. And the system should automatically record the basic information that is not returned to the guest, such as name , address, phone number, credit card, passport number, etc.

When the guest checks in, the system will assign the room to him or her until the guest checks out. The system must keep track of the guest's account, and print his or her bill. To avoid guests standing in a line while waiting for the Checkout, the all the procedure must take less than 60 seconds to complete.

1.3 General knowledge about the domain

Our client/stakeholder is Linnaeus Hotel. Our customers have already ordered a system program to replace the legacy system to enter the reservation and check in and out of the hotel. The new system can save money and help them better serve their guests.

1.4 Analysis the legacy system

Regarding the legacy system, we first think that the legacy system does not have an architectural model. The structure is very confusing and reading code is very inconvenient. Because there is no architectural pattern, we have found that the code is very messy to write together, some methods and data written in the same class, which is very unreasonable. Because these important data are not encapsulated, if you encapsulate this data, then users who use this class cannot easily manipulate the data structure directly, but can only execute data that the class allows to expose. This can avoid the external influence. On the internal data and improve the maintainability of the program.

Second, because the legacy system does not have an architectural model, it is difficult for us to modify the legacy system. This means that legacy system do not consider future updates. This is very detrimental to the program's development and updateability.

Third, when we run code, the program reacts very slowly because the legacy system writes methods, data, and GUI together, causing many methods and data not to be reused. Failure to increase the reusability of programs is a failure of the legacy system.

Finally, we think that the legacy system's GUI is very complex and unfriendly. And, because of the method, the data and the GUI cannot be separated, resulting in a GUI interface that cannot be easily modified.

1.5 Environment

The environment used will be a PC running Windows on which the application will run. System programs will be written and developed in Java. This makes it versatile and works on all the required platforms. We have received a legacy system that is written in Java and platform independent. It will run on any platform with a Java-VM. So our new system will also meet this environment requirement.

1.6 Tasks and procedures currently performed

About the legacy system that we received, I listed its main four functions.

Booking(Reservation):Whenever visitors want to book a room, they must fill in personal information. Also ask the guests what type of room and request they want.

Check in:When the guest wants to check in, The customer needs to tell the staff his information, then the staff enters the customer's information in the system and searches for their room information, and then let the guest check in.

Check out:When the guest wants to check out, The customer needs to inform the staff member of his information or room number. The staff then enters the customer's information or room number in the system and proceeds to find their room information., and then help the guest to check out.

Payment:After you have successfully checked out, the system will keep the visitor's personal information and print out the bill to the guest.

//quickly verify availability and how to find a room that meets your requirements if required. How to track guest accounts and quickly print the bill. How to record each room information and guest information. How to connect with the finance department.

1.7 Competing software

I searched the Internet for a system called OTA Hotel Management. This system is also designed for small and medium-sized hotels.
This system has a Real-time online booking engine, and can also be scheduled using tools such as Skype. And this system supports multiple languages. These advantages are not available in our program. I think we also need add function of secured login page and supports password encryption. It can protect the personal information of the guests and is not easily revealed.

1.8 Stakeholder’s vision for the future

“Linnaeus Hotel”, which manages 2 different buildings, 1 within the Växjö Campus and 1 within the Kalmar Campus(Therefore, we need to store information through the sql database so that the information in the two places can communicate with each other).Each building contains rooms in which guests can stay. Some hotel rooms adjoin others; there are internal doors between them. Each hotel room is assigned a quality level(e.g. a larger room or a room with a view would be better than a smaller room without a view). Each room also has a certain number and type of beds, a room number, and a smoking/non-smoking status. Each quality level has a maximum daily rate.

Requirements Document

1.Design the existing front desk system

The system should identify the number and type of each room. When the hotel staff entered the necessary information, the system must be validated in 2 seconds. And the system should automatically record the basic information that is not returned to the guest, such as name, address, phone number, credit card, passport number, etc.

When the guest checks in, the system will assign the room to him or her until the guest checks out. The system must keep track of the guest's account and print his or her bill. To avoid guests standing in a line while waiting for the checkout, the all the procedure must take less than 60 seconds to complete.

2.Functional requirements

FR1 The system manages multiple rooms

FR 1.1 several rooms can be created the room include room number, room type, view, floor, location(vaxjo/kalmar), adjoin room, smoking and animal, the room number identifies the room.

FR 1.2 each room can be deleted.

FR 1.3 each attributes of a room can be modified.

FR 2 The system manages multiple customers

FR 2.1 several customers can be added
include: id type, id, name, company, address, tel, email address, arrival time, departure time, room, days, card, price, smoker

FR 2.2Manage can change room and room price for each customer.

FR 3 reservation can be canceled at the anytime, but some fees may be charged

FR 4 The system be able search the information about the room
search with room type, arrival time, departure time and location
search with room number

FR 5 The system be able search the information about the guest
search with room number
search with name
search with arrival time
search with departure time
search with arrival time and departure time

FR 6 The system must keep track of the guest’s account, and print customer bill.

FR 7 The system has to show the rooms information

FR 7.1 All rooms have to be display

FR 7.2 After searching, the system should be able to display those rooms

FR 8 The system has to show guests information(name, room, arrival time, departure time, location(vaxjo/kalmar), arrived(true/false))

FR 8.1 All guests have to be display

FR 8.2 After searching, the system should be able to display those guests

3.Non-Functional requirements

NFR 1 Performance Requirements

NFR 1.1 When the system check the free room, this progress must take less than 2 seconds

NFR 1.2 Check out procedure must take in average less than 60 seconds

NFR 2 For the security, the system should have login page

NFR 2.1 Hotel staff will be able to log in to the Linnaeus Hotel Management system
NFR 3 The system has to have an user-friendly interface

4.Scenarios

These scenarios are based on the meeting we had with our customer.

4.1 Scenario for reservation

Steven wants to reservation. He walks to the front desk and tells Bob(the staff) about his personal information, room type and room requirements. After Bob enters this information, the system will check to see if there is an empty room that meets the Steven's needs. If there is, Bob will tell Steven the price. If the Steven has no opinion, he will make a reservation.

4.2 Scenario for check in

Steve wants to check in. He walks to the front desk and tells Bob(the staff) about his personal information. After Bob enters this information, the system will find the room that the customer has booked. After successfully finding the corresponding room number, Bob will tell the Steven the price. When Steven successfully pays, Bob will help the customer to check in.

4.2.1 Scenario for cancel the reservation

#1 Andy come to check in, but he changed his mind, wants to cancel the reservation. He walks to the front desk and tells Peter(the staff) about his personal information. After Peter enters this information, the system will find the room that Andy has booked. And Peter check the information that there has no extra fees, then cancel successful.

#2 Amy come to check in, but she changed his mind,wants to cancel the reservation. She walks to the front desk and tells Bob(the staff) about his personal information. After Bob enters this information, the system will find the room that Amy has booked. And Bob check the information and find out it cancel too late. Bob will told Amy that because you cancel too late, so you need to pay some fees. Amy pay the fees, cancel successful.

4.3 Scenario for check out

James wants to check out. He walks to the front desk and tells Kobe(the staff) about his personal information or room number. After Kobe enters this information, the system successfully finding the corresponding room number, Kobe will help James check out and print the bill.

4.4 Scenario for manage the room

#1.(add room)The hotel was expanded and Curry(the hotel staff) wanted to add rooms through the system. Curry successfully found the room number to be added and filled in the basic information of the room. Then Curry successfully added the room using the hotel system.

#2.(edit room)Kevin(the hotel staff) found an error in the information of the original room. He successfully used the system to find the room number. Then Kevin modified the room through the system.

#3.(delete room)Lebron(the hotel staff) found a room number that did not exist. He used the system to successfully find the room number. Then Lebron deleted the room through the system.

5.Use Cases

The following use cases are derived from the scenario.

5.1Use Case 1

Use Case 1 - Enter reservation

Primary Actor

Hotel staff : Operate and manage Linnaeus hotel reception system
Customers: To provide information

Goals: Hotel staff will help customers to make a reservation at system

Scope: Linnaeus hotel front-desk system

Preconditions

1.Customers want to make a reservation

Steps

1.Linnaeus hotel front-desk system has been started
2.The system will record the customer's basic information
3.The system will check the room type that the customer wants to check in.
4.The system will show the price of the room(staff need to fill in price).
5.Check if there is room available for guest.
6.After booking, the system shows that the guest's room is occupied.

Post condition

1.The use case terminates when the customers successful reservation.
2.The selected room is now unavailable to any other customers.

Exceptions

1.The hotel does not have the requested room type
2.There is no room for customers

5.2Use Case 2

Use Case 2 - Check in at system

Primary Actor

Hotel staff : Operate and manage Linnaeus hotel reception system
Customers: To provide information

Goals: Hotel staff will help customers to check-in at system

Scope: Linnaeus hotel front-desk system

Preconditions

1. Customers already made a reservation
2. Customers want to check in

Steps

1. Linnaeus hotel front-desk system has been started
2. Hotel staff enter the customer's information, let system search for it.
3. The system will find the room that customer has booked

4. Check if there is room available for guest.
5. The system shows that the customer's room is occupied.
6. Check-in successful

Post condition
1.The use case terminates when the customer successful check-in.
2.The selected room is now unavailable to any other customers.
3.The system records information about customers and their rooms in a list.

Exceptions

1. The system cannot find customer’s information.
2. There is no room for customer.
3. The customer doesn't want to check in.

5.2.1 Use Case 2.1

Use Case 2.1 - Cancel the reservation

Primary Actor
Hotel staff : Operate and manage Linnaeus hotel reception system
Customers: To provide information

Goals: Hotel staff will help customers to cancel the reservation at system.

Scope: Linnaeus hotel front-desk system

Preconditions
1.Customers already made a reservation.
2.Customers don’t want to check in.

Steps
1.Linnaeus hotel front-desk system has been started
2.Hotel staff enter the customer’s information,let system search for it.
3.The system will find the room that customer has booked
4.Hotel staff help customers to cancel the reservation.

5.The system shows that the room is available.
6.After Cancel, check out if there is any extra charge.

Post condition
1.The use case terminates when the customer successful cancel the reservation.
2.The room is now available to any other customers.

Exceptions

1.The system could not find the room through information.
2.The system was unable to record additional costs.
3.The customer has no extra money to pay the extra fee.

5.3Use Case 3

Use Case 3 - Check out at system

Primary Actor
Hotel staff : Operate and manage Linnaeus hotel reception system
Customers: To provide information

Goals: Hotel staff will help customers to check-out at system.

Scope: Linnaeus hotel front-desk system

Preconditions
1.The scheduled check-out date was arrived.
2.Customers want to check out.

Steps

1.Linnaeus hotel front-desk system has been started
2.The system finds the room through information.

3.The system records the guest's account.
4.After Check-out, the system will print the guest's bill.

Post condition
The use case terminates when the customers successful check-out.
The room is now available to any other customers.

Exceptions

1.The system could not find the room through information.
2.The customer's account cannot be recorded by the system.
3.The system cannot print the bill.

5.4 Use Case 4

Use Case 4 - Manage the room

Primary Actor
Hotel staff : Operate and manage Linnaeus hotel reception system

Goals: Hotel staff will manage rooms at system.

Scope: Linnaeus hotel front-desk system

Preconditions

1.The hotel was renovated.
2.Hotel staff want to manage the room.

Steps for add rooms:
1.Linnaeus hotel front-desk system has been started.
2.Hotel staff use system and enter the management page.
3.Hotel staff will use the system to add the room.

Steps for edit rooms:

1.Linnaeus hotel front-desk system has been started.
2.Hotel staff use system and enter the management page.
3.Hotel staff will use the system to edit the room's information.

Steps for delete rooms:

1.Linnaeus hotel front-desk system has been started.
2.Hotel staff use system and enter the management page.
3.Hotel staff will use the system to delete the room.

Post condition

1.The use case terminates when the customer successful manage rooms.
2.After management, these rooms will be redisplayed in the list.

Exceptions

1.The hotel does not have an extension and does not need to add new rooms
2.There is no problem with the hotel's room information
3.Hotel does not need to delete room

6.Use Case Diagram

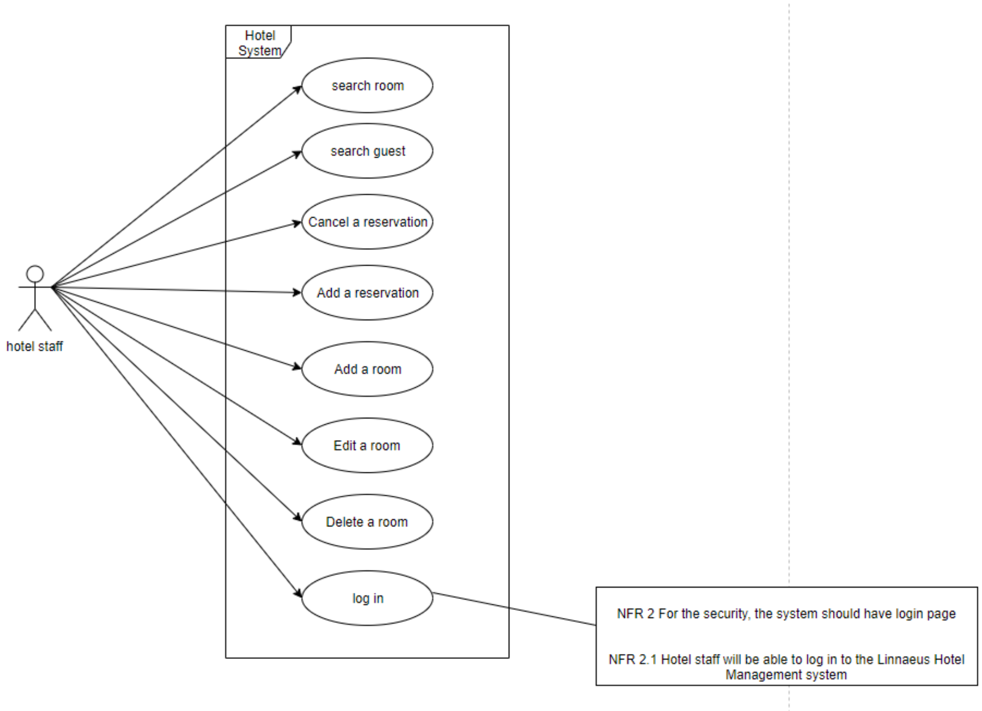
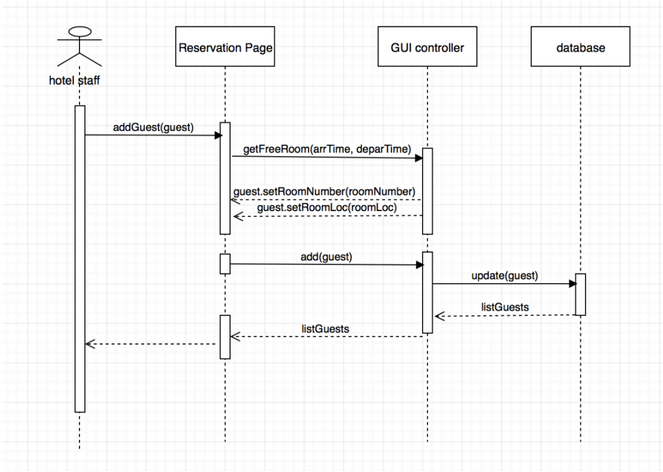

Hotel System
search room
search guest
Cancel a reservation
Add a reservation
Add a room
Edit a room
Delete a room
log in
hotel staff
NFR 2 For the security, the system should have login page
NFR 2.1 Hotel staff will be able to log in to the Linnaeus Hotel Management system

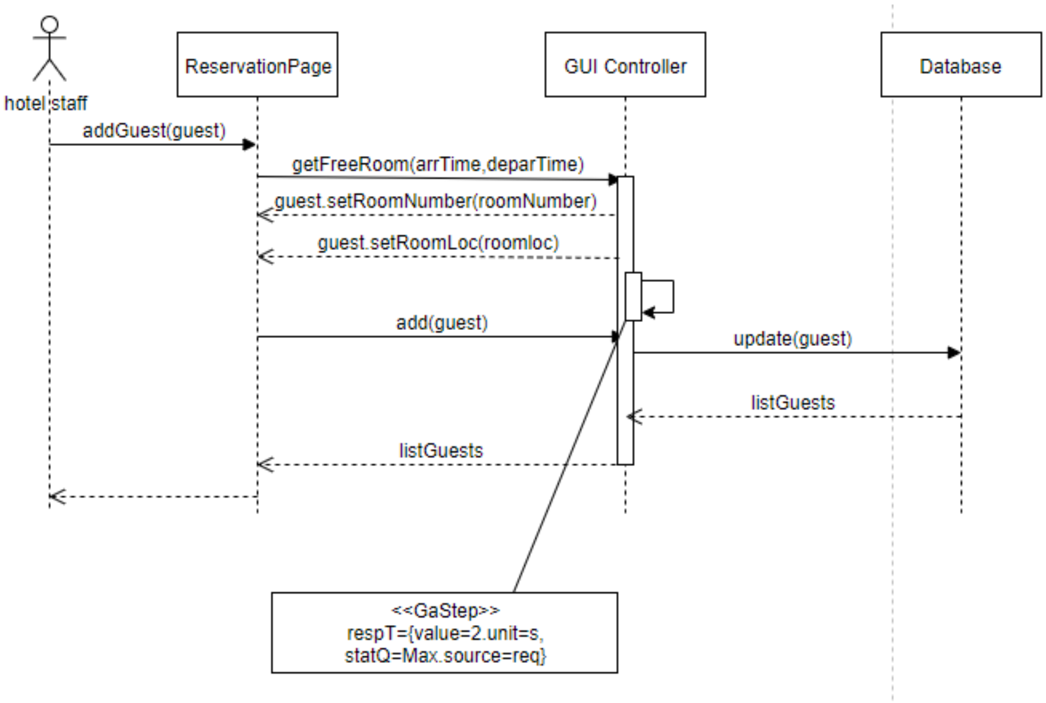
Figure 5.1: Use Case

7.Sequence Diagram

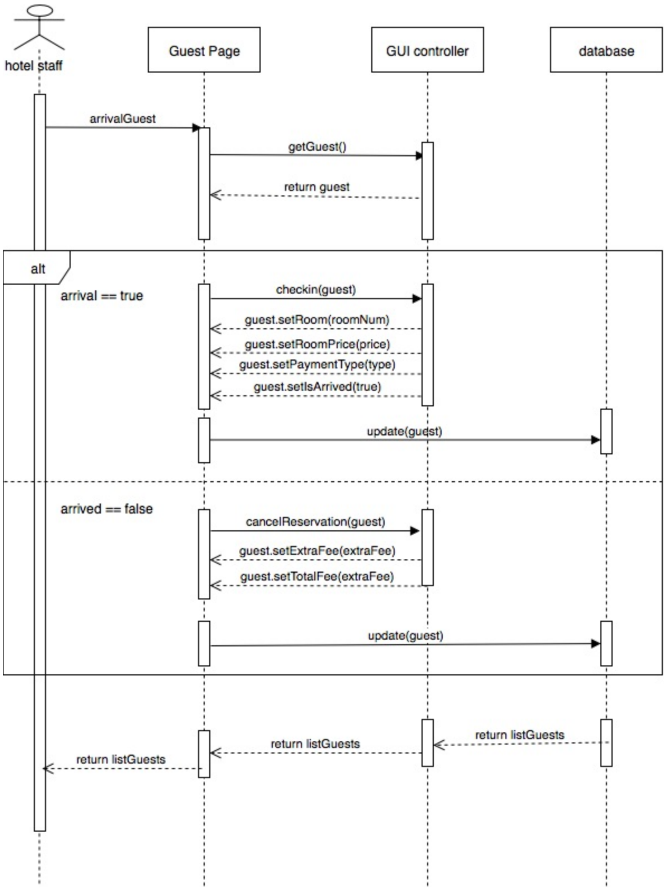
Reservation Sequence diagram



NFR 1.1 When the system check the free room, this progress must take less than 2 seconds for MARTE

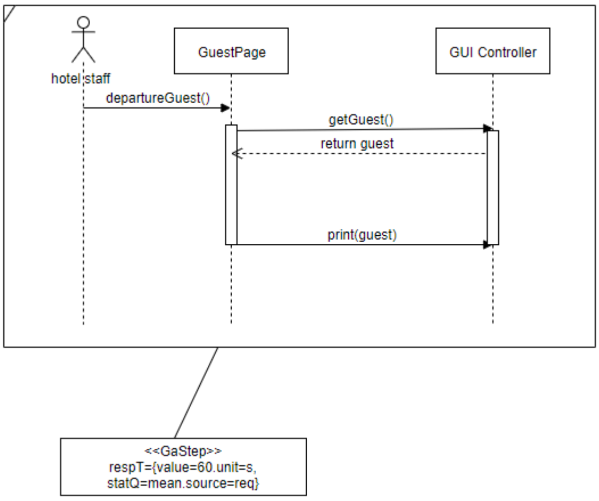


Check-in Sequence diagram

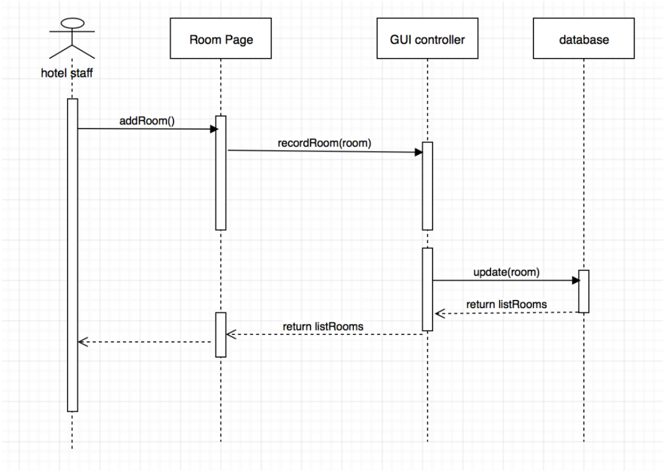


Check-out Sequence diagram

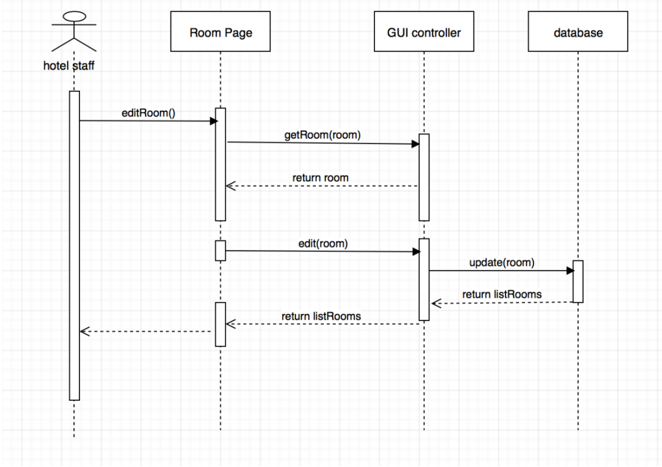
NFR 1.2 Check out procedure must take in average less than 60 seconds for MARTE



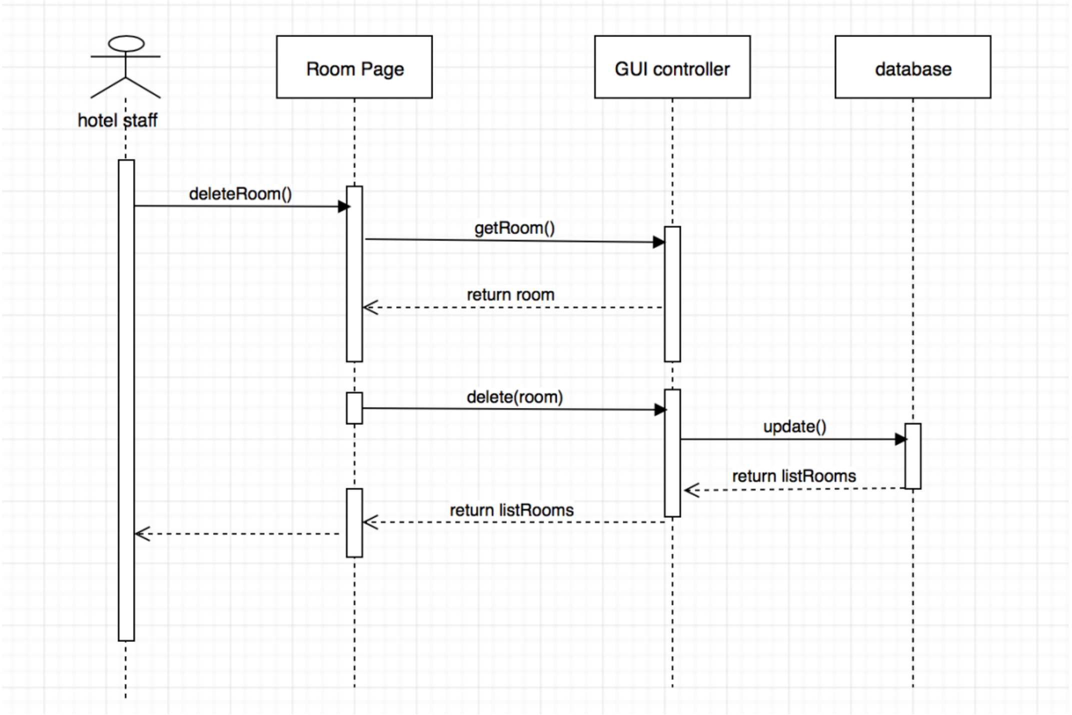
Add Room Sequence diagram



Edit Room Sequence diagram

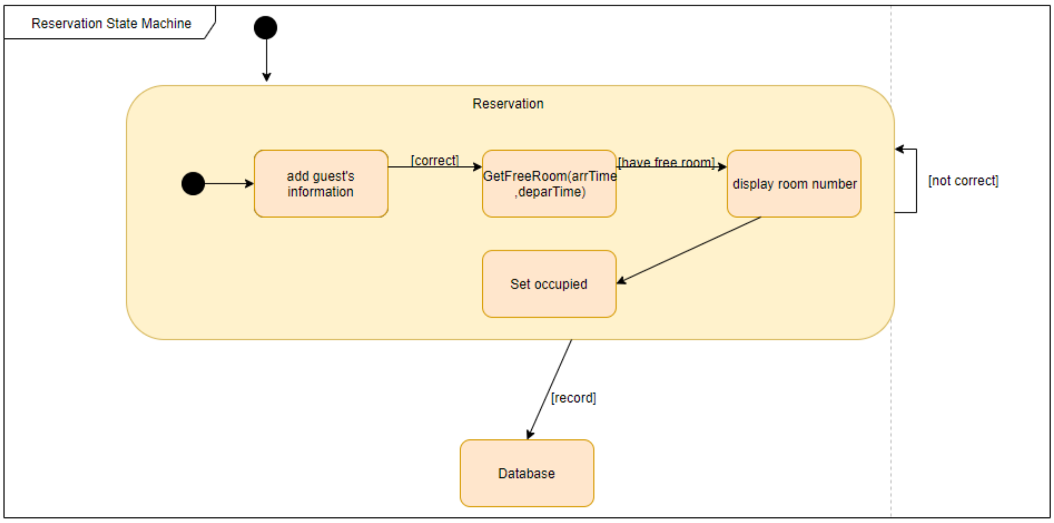


Delete Room Sequence diagram

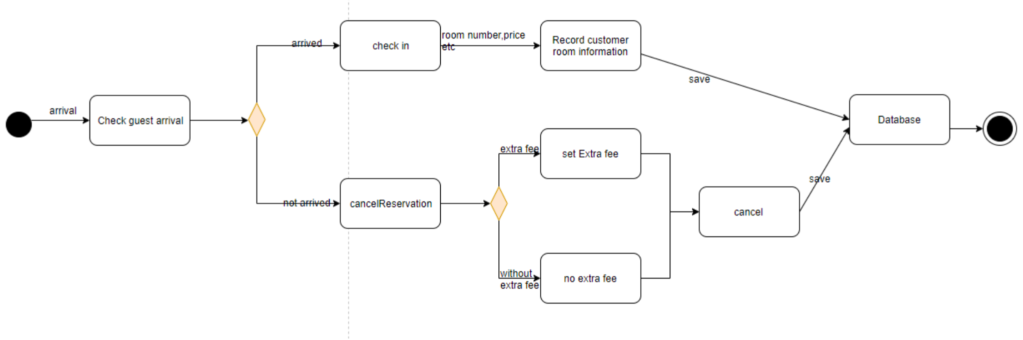


8.State machine diagram

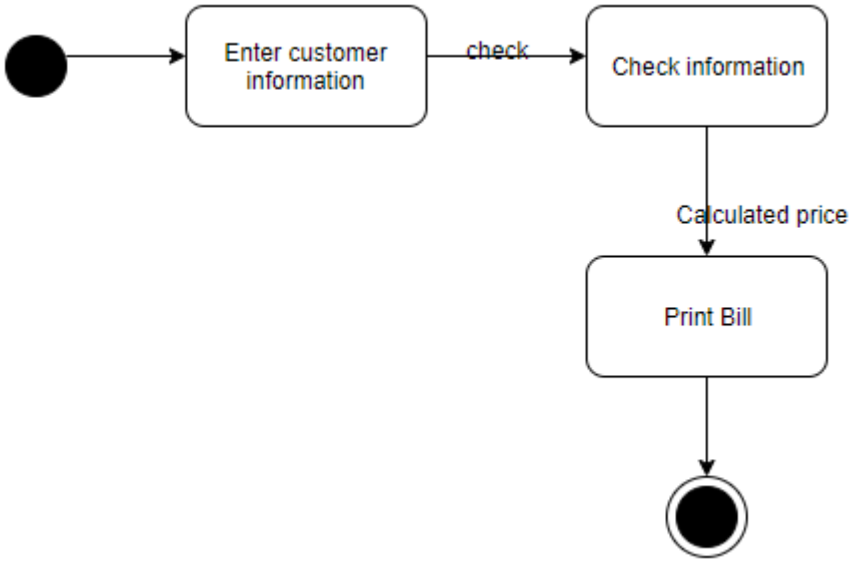
State machine diagram for Reservation



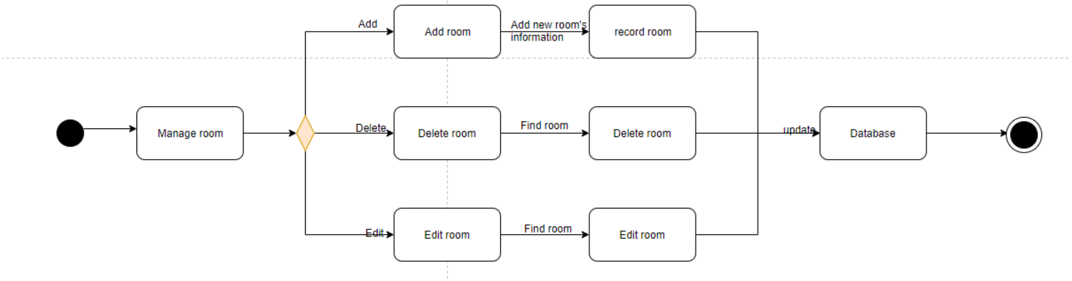
State machine diagram for Check-in and Cancel reservation



State machine diagram for Check-out

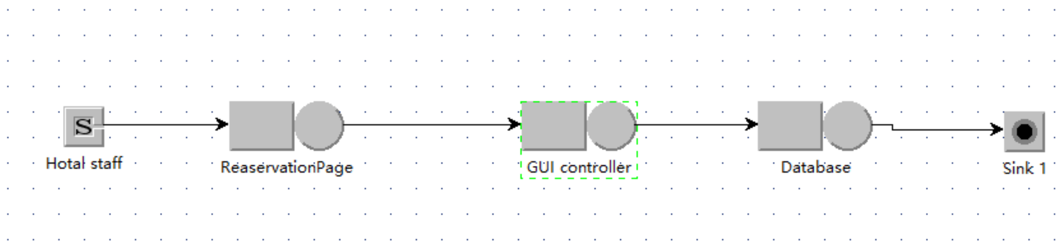


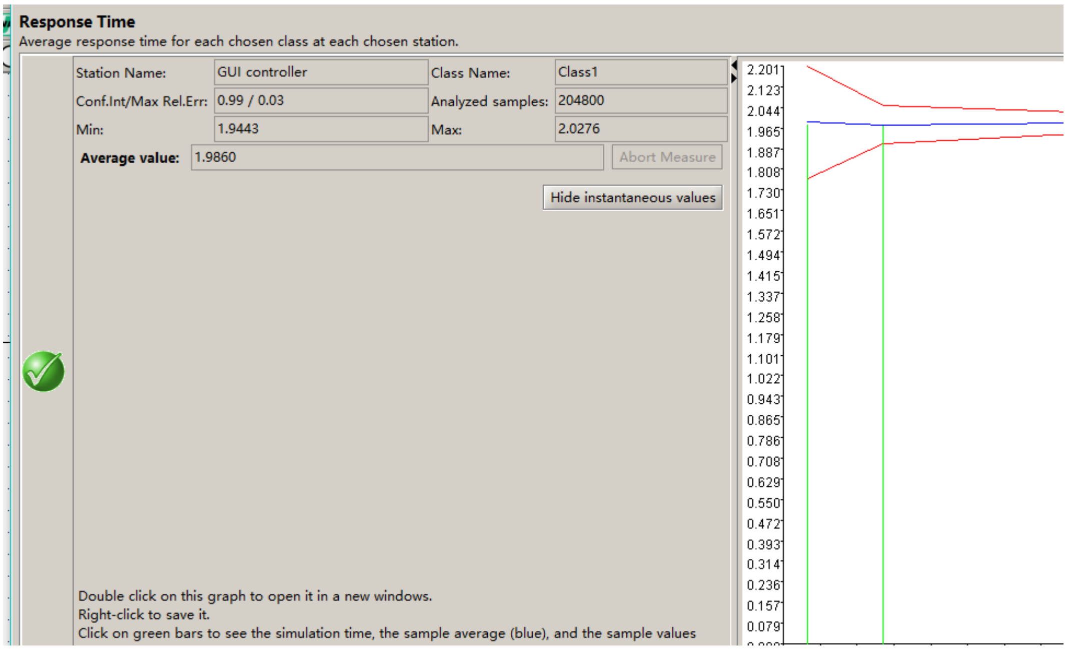
State machine diagram for Manage room



9. Performance Evaluation

Performance Evaluation for **NFR 1.1 When the system check the free room, this progress must take less than 2 seconds**



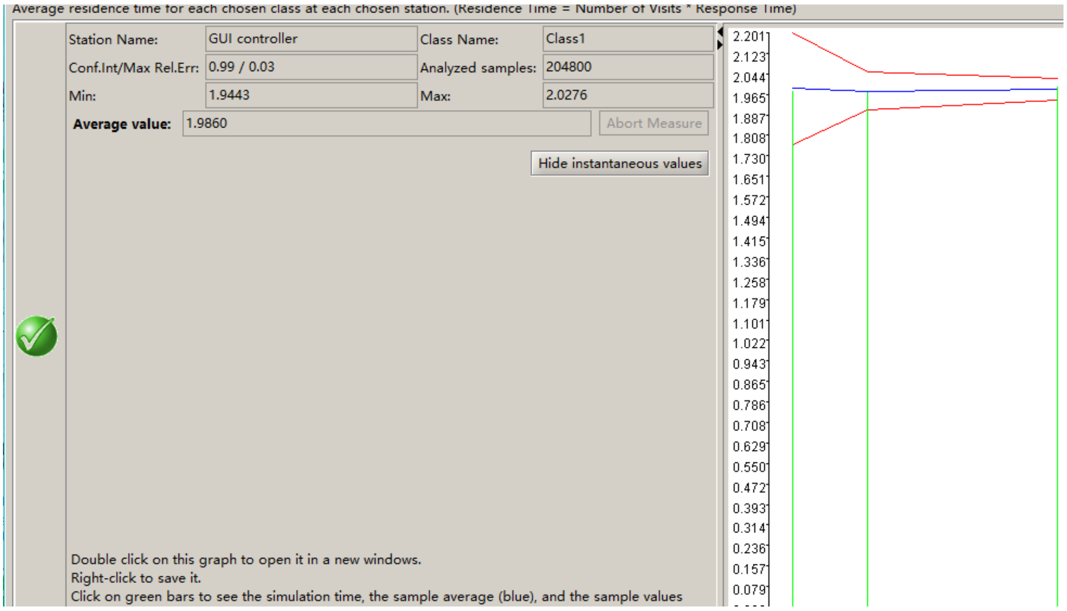


Response Time
Average response time for each chosen class at each chosen station.

Station Name: GUI controller Class Name: Class1
Conf.Int/Max Rel.Err: 0.99 / 0.03 Analyzed samples: 204800
Min: 1.9443 Max: 2.0276
Average value: 1.9860 Abort Measure

Hide instantaneous values

Double click on this graph to open it in a new windows.
Right-click to save it.
Click on green bars to see the simulation time, the sample average (blue), and the sample values



Average residence time for each chosen class at each chosen station. (Residence Time = Number of Visits * Response Time)

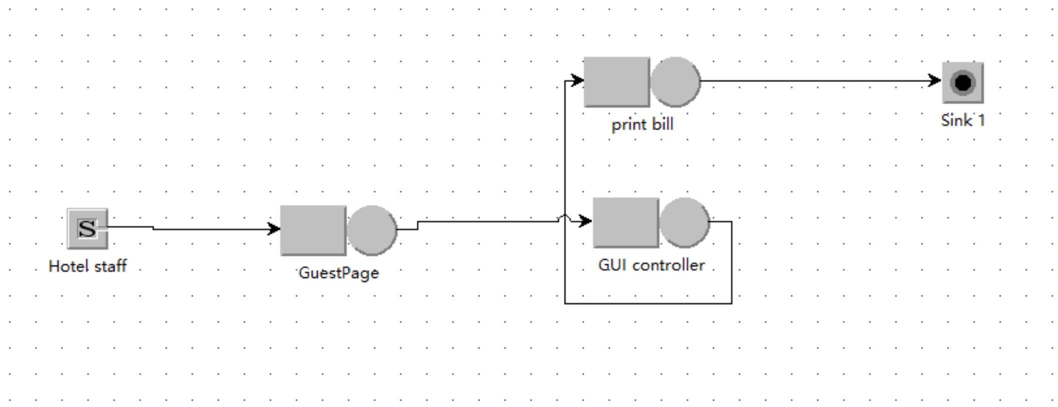
Station Name: GUI controller Class Name: Class1
Conf.Int/Max Rel.Err: 0.99 / 0.03 Analyzed samples: 204800
Min: 1.9443 Max: 2.0276
Average value: 1.9860 Abort Measure

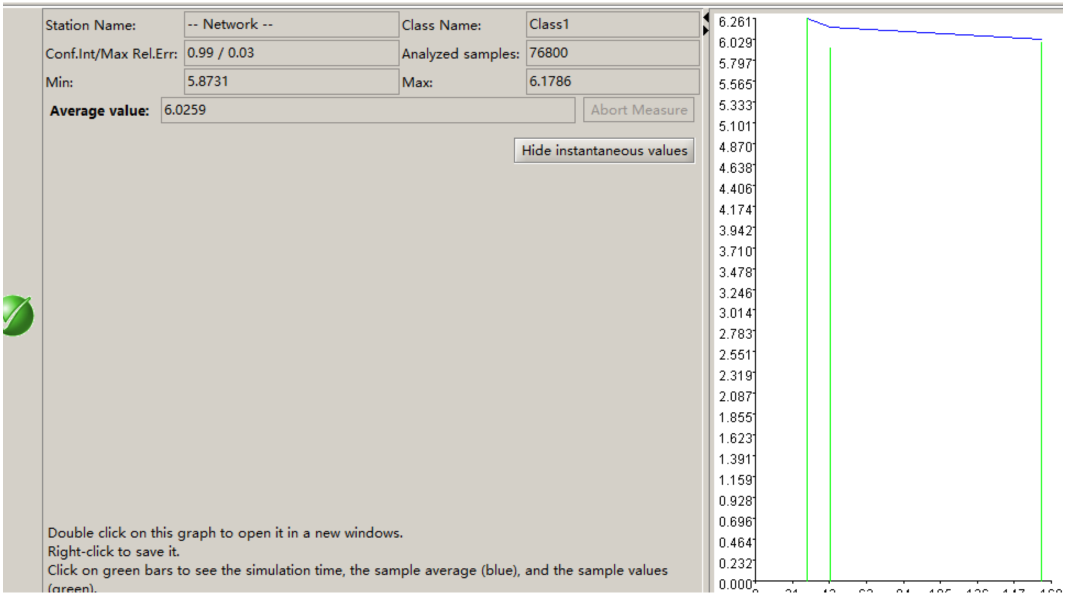
Hide instantaneous values

Double click on this graph to open it in a new windows.
Right-click to save it.
Click on green bars to see the simulation time, the sample average (blue), and the sample values

We tested the response time and wait time of the program after entering the customer's personal information.

Performance Evaluation for **NFR 1.2 Check out procedure must take in average less than 60 seconds**




Station Name: -- Network --
Class Name: Class1
Conf.Int/Max Rel.Err: 0.99 / 0.03
Analyzed samples: 76800
Min: 5.8731
Max: 6.1786
Average value: 6.0259

Double click on this graph to open it in a new windows.
Right-click to save it.
Click on green bars to see the simulation time, the sample average (blue), and the sample values (green).

We tested the entire check-out process, from finding customers to printing bills. Because of the simplicity of our program, we use it for a shorter time.

10.narrow the problem statement

In the first version, I don't think we need a guest list who cancel the reservation and pay extra, because I don't think it's important for the first release.There is also a time limit requirement, which I think can be improved in later versions.

11.TimeLog

Activities	Estimate	Use time	End Date
Domain Analysis Document	2 hours	1 hours 30 mins	2018/5/23
Requirements Document	8 hours 15 mins	6 hours 35 mins	2018/5/23
TimeLog	5min	5mins	2018/5/23
Use Case (include diagram)	3 hour	2 hours 15 mins	2018/5/23
State machine diagram	1 hour	1 hours	2018/5/23
Scenario	1 hour 30 mins	1 hour	2018/5/23
Functional requirement	20mins	20mins	2018/5/23
Non-Functional requirement	20mins	20mins	2018/5/23
Sequence Diagram	2 hours	1 hours 20 mins	2018/5/23
Others	30 mins	15 mins	2018/3/29