



SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

OODP UML Project based on Hotel Management System

Under the guidance of Dr. Alice nithya ,
Associate Professor

Completed by:

Shaik ismiya

RA2111047010148

Section c– AI

AIM:

To create a system to perform the Hotel Management System.

PROBLEM STATEMENT:

A Hotel System manages information about rooms, reservations, customers, and customer billing. A customer can make reservations, change, or cancel reservations through the hotel website. When a customer makes reservations, he/she needs to check if a room the customer wants to reserve is available. If a room is available, the customer enters his/her information to the system and receives a confirmation number from the web site.

A desk clerk checks in a customer with only a prior reservation, change the checkout date, and checks out the customer. A room is assigned to the customer at check-in time and a customer billing record is created at that time. The customer billing record is updated every night at 12. When a customer checks out, the desk clerk prints the bill. A customer can pay by cash, check, or credit card when he/she checks out.

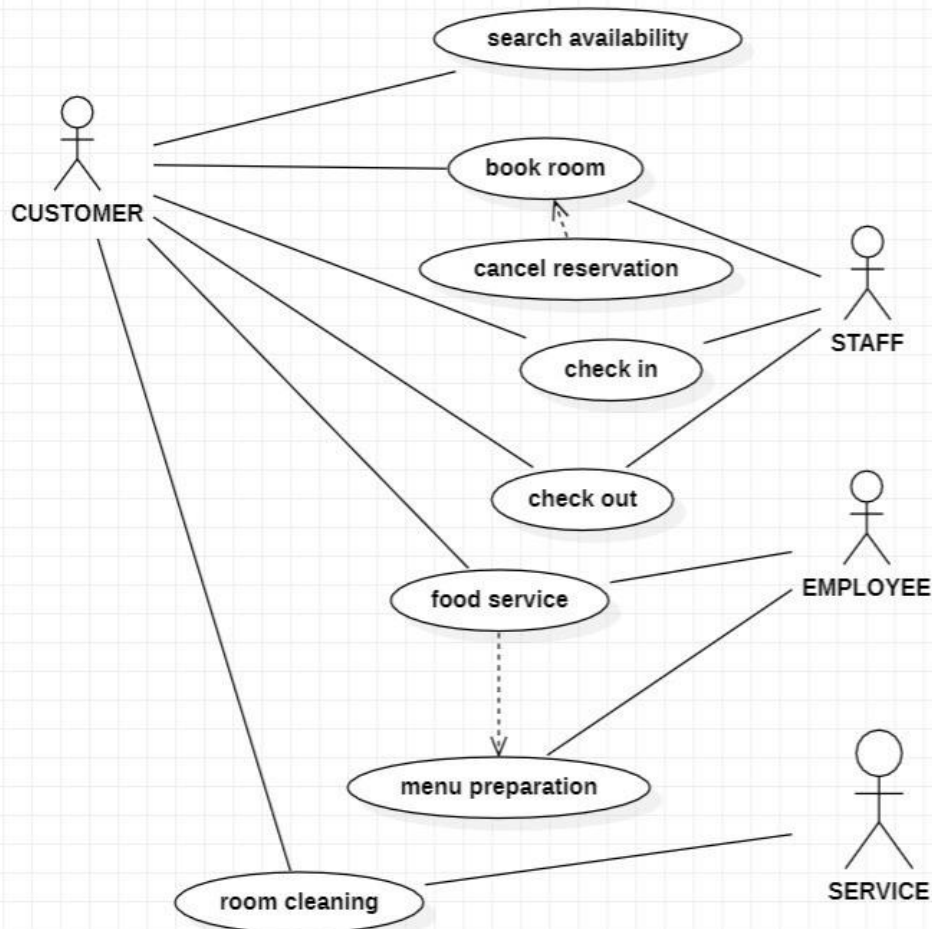
TECHNOLOGIES TO BE USED

- HTML • JSP
- Javascript
- Java
- C++

TOOLS TO BE USED

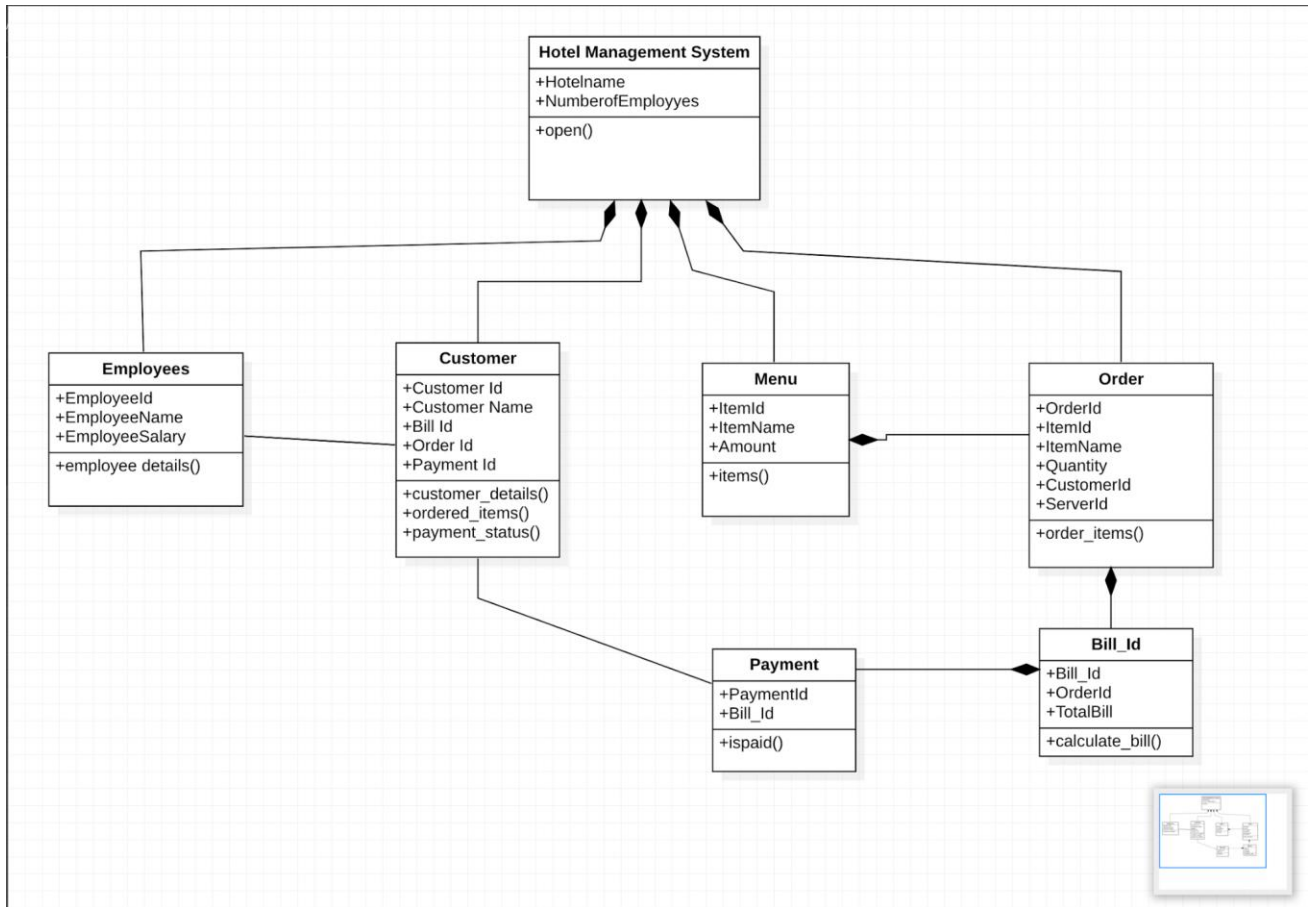
- CodeBlocks IDE (Integrated Development Environment)
- StarUML (for developing UML Patterns)

USE CASE DIAGRAM



This Use Case Diagram is a graphic depiction of the interactions among the elements of the Hotel Management System. It represents the methodology used in system analysis to identify, clarify, and organize system requirements of Hotel Management System. The above main actors of Hotel Management who perform the different types of use cases such as Manage Hotel, Manage Rooms, Manage Services, Manage Payments, Manage Booking, Manage Customers, Manage Users and Full Hotel Management System Operations.

CLASS DIAGRAM



CLASS DIAGRAM:

Hotel Management System Class Diagram describes the structure of a Hotel Management System class , their attributes, operations (or methods), and the relationships among objects. The main classes of the Hotel Management System are Hotel, Rooms, Services, Payments, Booking, Customers .

Classes of Hotel Management System Class Diagram:

- **Hotel Class** : Manage all the operations of Hotel
- **Rooms Class** : Manage all the operations of Rooms
- **Services Class** : Manage all the operations of Services
- **Payments Class** : Manage all the operations of Payments
- **Booking Class** : Manage all the operations of Booking

- **Customers Class** : Manage all the operations of Customers

Generated Source Code →

Bill_Id –

/**

* Project Untitled

*/

#ifndef _BILL_ID_H

#define _BILL_ID_H

class Bill_Id {

public:

void Bill_Id;

void OrderId;

void TotalBill;

void calculate_bill();

};

#endif //_BILL_ID_H

Class1 –

/**

* Project Untitled

*/

#ifndef _CLASS1_H

#define _CLASS1_H

class Class1 {

};

#endif //_CLASS1_H

Customer –

/**

* Project Untitled

*/

#ifndef

_CUSTOMER_H

#define

_CUSTOMER_H class

Customer { public:

void Customer Id; void

Customer Name;

void Bill Id; void Order

Id; void Payment Id;

void

customer_details();

void ordered_items();

void payment_status();

};

#endif //_CUSTOMER_H

Employees –

/**

* Project Untitled

*/

#ifndef _EMPLOYEES_H

#define _EMPLOYEES_H

class Employees {

public:

```

void EmployeeId;
void EmployeeName;
void EmployeeSalary;
void employee details();
};
#endif //_EMPLOYEES_H

```

```

Hotel Management System –
/**
 * Project Untitled
 */
#ifndef _HOTEL MANAGEMENT
SYSTEM_H #define _HOTEL
MANAGEMENT SYSTEM_H class Hotel
Management System { public:
void Hotelname; void
NumberofEmployyes;
void open();
};
#endif //_HOTEL MANAGEMENT SYSTEM_H

```

```

Interface1 –
/**
 * Project Untitled
 */

#ifndef _INTERFACE1_H
#define _INTERFACE1_H
class Interface1 {
};

```

```
#endif //_INTERFACE1_H
```

Menu –

```
/**
```

```
* Project Untitled
```

```
*/
```

```
#ifndef _MENU_H
```

```
#define _MENU_H
```

```
class Menu {
```

```
public: void
```

```
ItemId; void
```

```
ItemName; void
```

```
Amount; void
```

```
items();
```

```
};
```

```
#endif //_MENU_H
```

Order –

```
/**
```

```
* Project Untitled
```

```
*/
```

```
#ifndef _ORDER_H
```

```
#define _ORDER_H
```

```
class Order {
```

```
public: void
```

```
OrderId; void
```

```
ItemId; void
```

```
ItemName; void
```

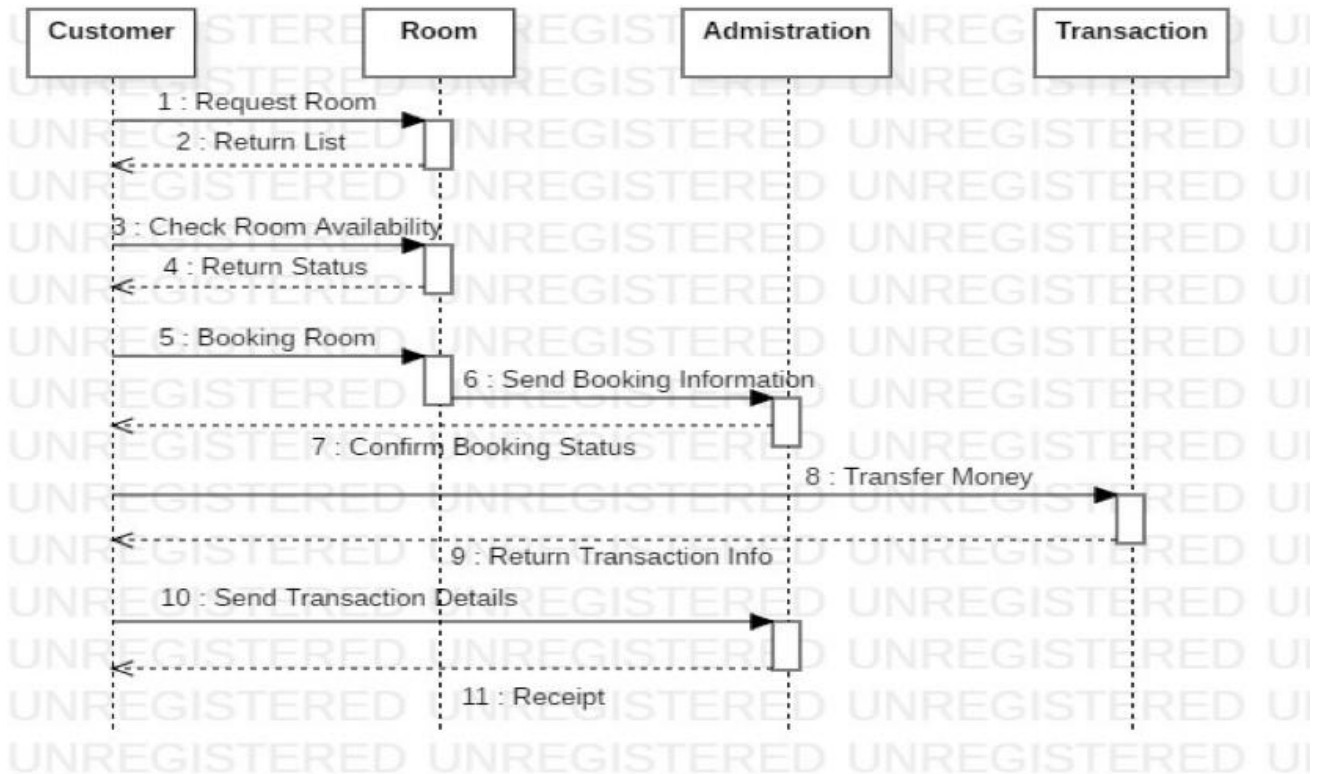
```
Quantity; void
```



```
CustomerId;  
void ServerId;  
void order_items();  
};  
#endif //_ORDER_H
```

```
Payment –  
/**  
 * Project Untitled  
 */  
#ifndef  
_PAYMENT_H  
#define  
_PAYMENT_H class  
Payment { public:  
void PaymentId; void  
Bill_Id; void ispaid();  
};  
#endif //_PAYMENT_H
```

SEQUENCE DIAGRAM



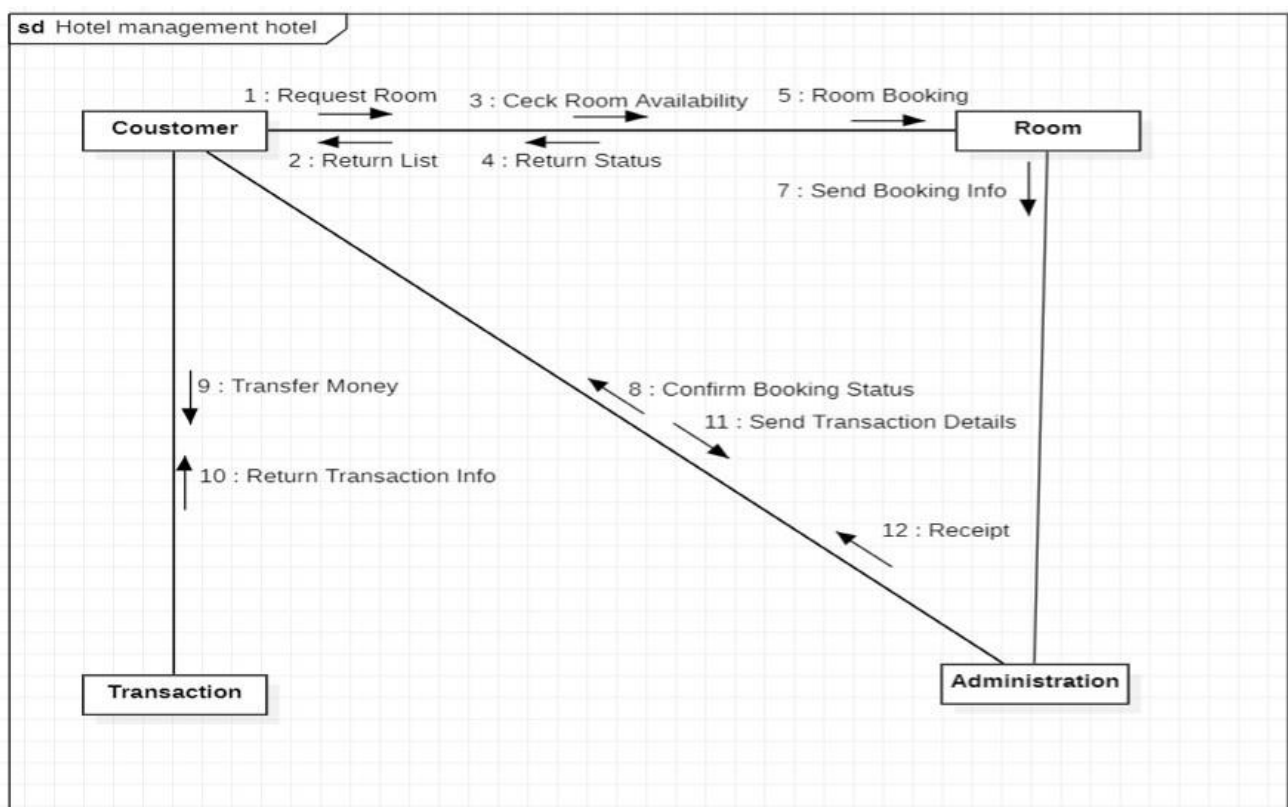
SEQUENCE DIAGRAM:

- A sequence diagram represents the sequence and interactions of a given USE-CASE or scenario. Sequence diagrams can capture most of the information about the system.
- Most object to object interactions and operations are considered events and events include signals, inputs, decisions, interrupts, transitions and actions to or from users or external devices.
- An event also is considered to be any action by an object that sends information.
- The event line represents a message sent from one object to another, in which the “form” object is requesting an operation be

performed by the “to” object. The “to” object performs the operation using a method that the class contains.

- It is also represented by the order in which things occur and how the objects in the system send message to one another.
- The sequence diagram for each USE-CASE that exists when a user administrator, check status and new registration about passport automation system are given.

COLLABORATION DIAGRAM



COLLABORATION DIAGRAM:

- It illustrates object interaction in a graph or network format in which objects can be placed anywhere on the diagram.
- In a collaboration diagram the relationship between applicant, system admin and authority. The people must fill the application form according

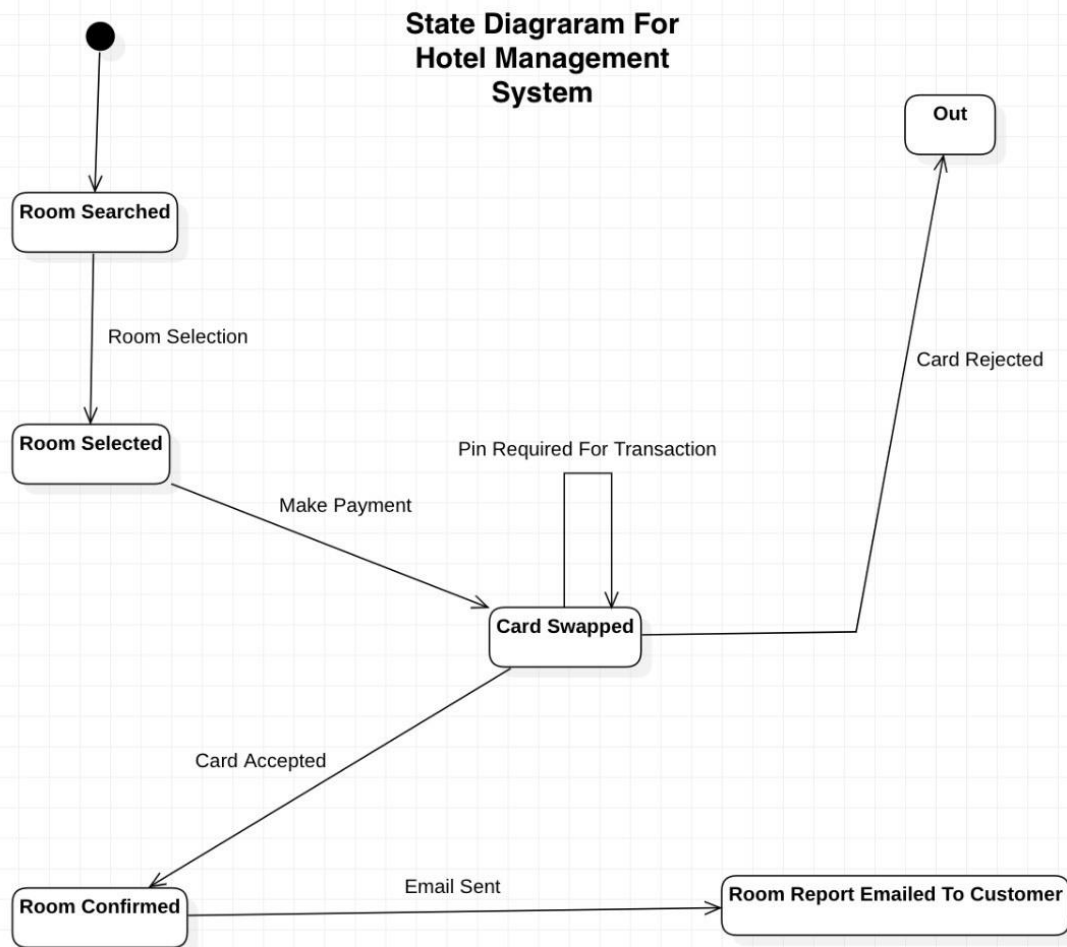
to detailed furnishings. The system admin verifies all details and forward details to authority. He enquiry the person and then issues the passport to the applicant.

Communication diagrams offer benefits similar to sequence diagrams, but they will offer a better understanding of how components communicate and interact with each other rather than solely emphasizing the sequence of events.

They can be a useful reference for businesses, organizations, and engineers who need to visualize and understand the physical communications within a program.

- Model the logic of a sophisticated procedure, function, or operation.
- Identify how commands are sent and received between objects or components of a process.
- Visualize the consequences of specific interactions between various components in a process.
- Plan and understand the detailed functionality of an existing or future scenario.

STATECHART DIAGRAM

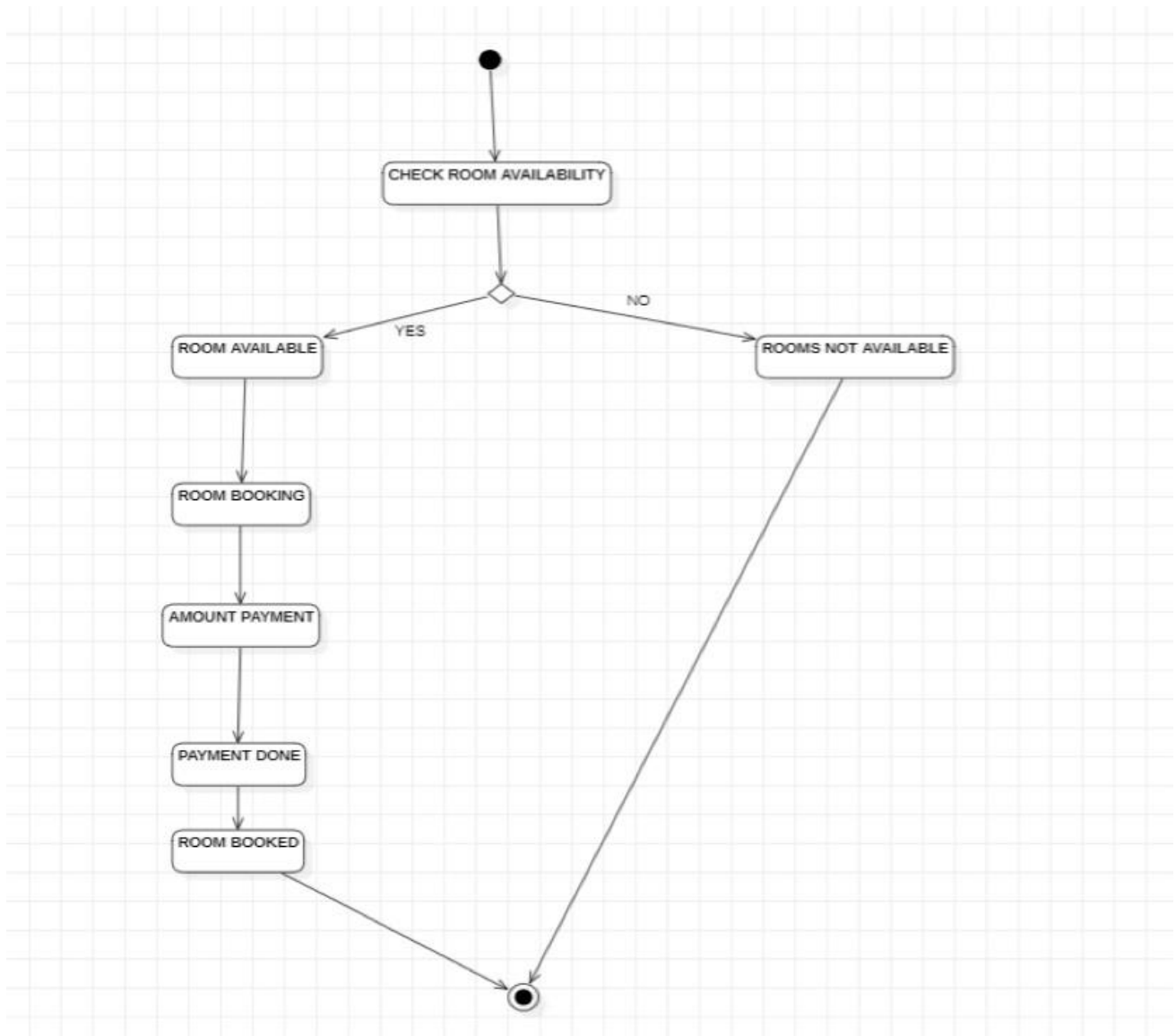


STATE CHART DIAGRAM:

This is **State chart diagram** of **Hotel management system** where customer will visit the rooms of the hotel and check whether the rooms are available or not if the rooms are available then customers go to the administrator and tell them to confirm the booking of their rooms after that customer pays the rent for rooms and take the receipt from the administrator

This is the **UML state chart diagram** of the Hotel management system which shows the interaction between **initial,final,state,choice,transition,and self transition.**

ACTIVITY DIAGRAM

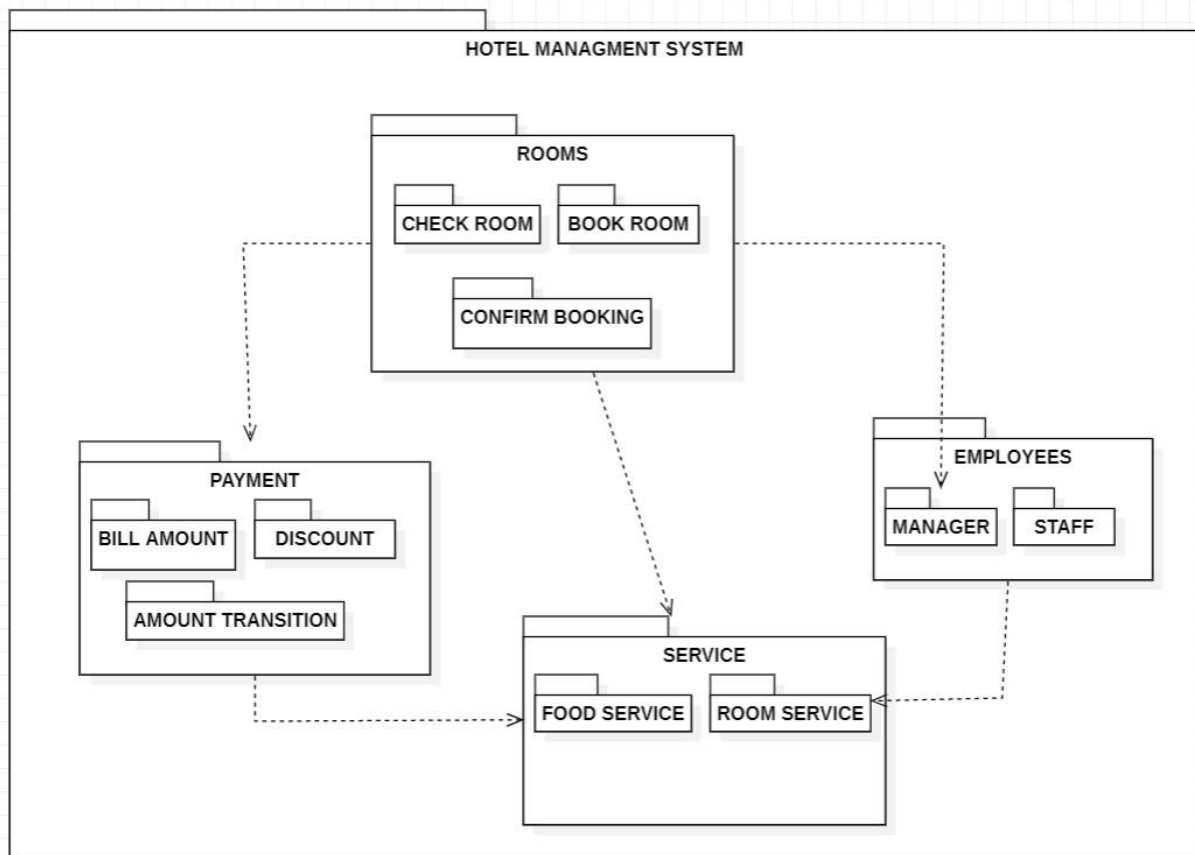


ACTIVITY DIAGRAM:

This is Activity diagram of Hotel management system where customer will visit the rooms of the hotel and check whether the rooms are available or not if the rooms are available then customers go to the administrator and tell them to confirm the booking of their rooms after that customer pays the rent for rooms and take the receipt from the administrator.

This is the **UML Activity diagram** of a Hotel management system which shows the interaction between **initial,final,state,choice,transition,self transition**.

PACKAGE DIAGRAM



PACKAGE DIAGRAMS :

Structural diagrams used to show the organization and arrangement of various model elements in the form of packages.

A package is a grouping of related UML elements , such as diagrams, documents, classes, or even other packages. Each element is nested within the package, which is depicted as a file folder within the diagram, then arranged hierarchically within the diagram.

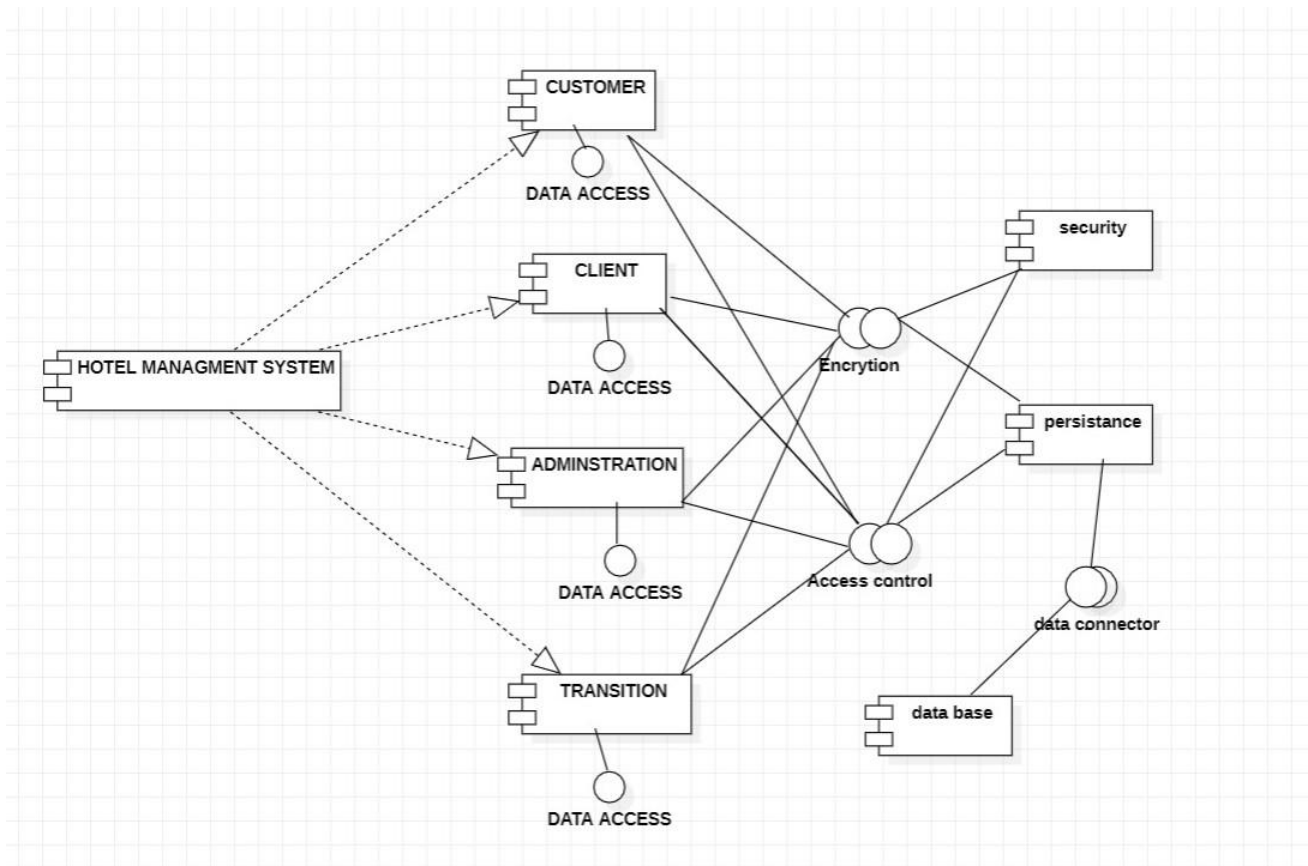
Package diagrams are most commonly used to provide a visual organization of the layered architecture within any UML classifier, such as a software system.

Benefits of a package diagram

A well-designed package diagram provides numerous benefits to those looking to create a visualization of their UML system or project.

- They provide a clear view of the hierarchical structure of the various UML elements within a given system.
- These diagrams can simplify complex class diagrams into well-ordered visuals.
- They offer valuable high-level visibility into large-scale projects and systems.
- Package diagrams can be used to visually clarify a wide variety of projects and systems.
- These visuals can be easily updated as systems and projects evolve.

COMPONENT DIAGRAM



COMPONENT DIAGRAM:

A component represents a modular part of a system, that encapsulates its contents and whose manifestation is replaced with in its environment. A component defines its behaviours in terms of provide and required interfaces.

A component diagram in UML gives a bird's-eye view of your software system. Understanding the exact service behaviour that each piece of your software provides will make you a better developer.

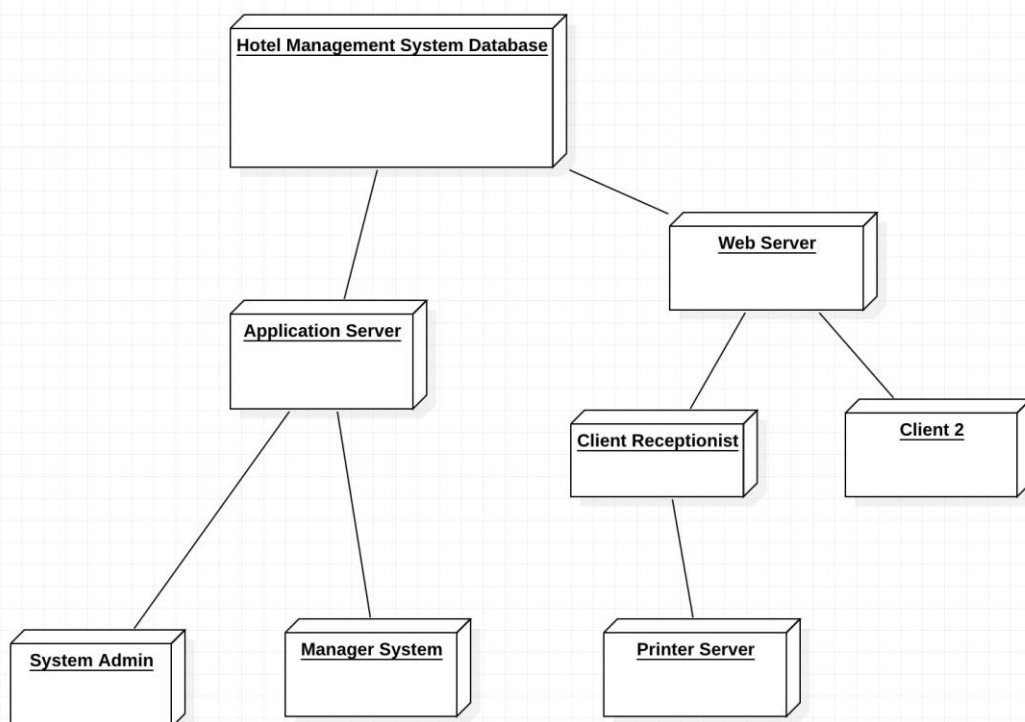
Component diagrams can describe software systems that are implemented in any programming language or style.

Benefits of component diagrams:

Though component diagrams may seem complex at first glance, they are invaluable when it comes to building your system. Component diagrams can help your team:

- Imagine the system's physical structure.
- Pay attention to the system's components and how they relate.
- Emphasize the service behaviour as it relates to the interface.

DEPLOYMENT DIAGRAM



Deployment Diagram For Hotel Management System

DEPLOYMENT DIAGRAM:

- A deployment shows the assignment of concrete artifacts to computational nodes. It shows the deployment of software elements to the physical architecture, and the communication. Deployment diagrams are useful to communicate the physical and deployment architecture.
- In the deployment diagram the object reference in component diagram is also included the deployment diagram. In this authority and system admin, interface through the people.
- It is the process of installing the program.

Deployment diagram applications:

Deployment diagrams have several valuable applications. You can use them to:

- Show which software elements are deployed by which hardware elements.
- Illustrate the runtime processing for hardware.
- Provide a view of the hardware system's topology.

CONCLUSION AND FUTURE SCOPE

Thus, the above project for the Hotel Management System has been successfully executed and codes are generated.

RESULT:

Thus the mini project for the Hotel Management System has been successfully executed and codes are generated.

REFERENCES:

www.google.com

<https://upload.wikimedia.org/wikipedia/en/f/fe/Srmseal.png>

https://sparxsystems.com/enterprise_architect_user_guide/15.2/model_domains/generatesourcecode.html#:~:text=Source%20code%20generation%20is%20the,elaborated%20with%20variables%20and%20methods .

<https://www.lucidchart.com/pages/tour>

<https://creately.com/blog/diagrams/uml-diagram-types-examples/>

https://www.tutorialspoint.com/uml/uml_standard_diagrams.htm

<https://www.smartdraw.com/uml-diagram/>