

**SE PRACTICAL**

**Student Details :-**

* Name :- Ankit Senjaliya
* Enrollment No. :- 19BT04046
* Branch :- Computer Science And Engineering
* Semester :- 5

**Faculty Mentor Details :-**

* Mr. Jagat Avashia

### Practical 1 :- Problem statement For Hotel Management System

**Problem Statement :-**

Web application for Hotel Management. take the selection criteria from user and display the hotels list for user basing on the criteria. User can book the room if there is availability of the rooms in that particular hotel. There are three different types of user roles for the application they are administrator, hotel agent and nor- mal user. Following are the actions provided for each user.

### Normal user :-

Can register for the site

Search the hotel details basing on the criteria. Book the hotel room

Can modify the self details.

### Hotel Agent :-

Can register for the site

Can add/update the details of the hotel.

### Administrator :-

Will approve the new hotel details added to the application Can delete the user/hotel details.

### Minimal Features :-

Adding Hotel information such as hotel name, location, number of rooms, facilities etc. to the database

Listing the hotels based on different criteria selected by the user. User able to select a hotel and book a room.

Booking permitted only if there are rooms available Displaying the reservation status.

Registration of users. Update user details. Modify hotel details.

Approval of the details entered by the hotel agent.

Delete user/hotel details by admin

### Additional Features :-

Taking Feedback from user.

Rating the hotels based on the feedback.

### Goals :-

To complete the hotel booking flow, registration of the users, adding details of the hotels.

### Deliverables :-

Minimal Features and additional features provided if time permits.

### Out of scope :-

We are not embedding payment gateway in the current application we will try if time permits after completion of additional features.

# Practical 3 :- Prepare Following UML Diagrams

# 1.) Draw Class Diagram

**Class Diagram For Hotel Management System**

### Class diagram is a type of UML diagram which shows the properties and relation- ships among various objects.

**Class :-**

**The classes used in this system are,**

**Hotel Management :-** This class depicts the entire hotel and says whether the hotel is opened or closed.

**Employees :-** It contains the details of the Employee. There are two kinds of employees, Server and the chef. This employee class is the parent class of two sub- class – Server and Chef

**Server :-** It contains the details of the server, the table to which they are as- signed, the order which is currently serving, etc.

**Chef :-** It contains the details of the chef working on a particular order.

**Customer :-** It contains the details of the customer.

**Table :-** It contains the table details like table number and the server who are as- signed to that table.

**Menu :-** Menu contains all the food items available in the restaurant, their avail- ability, prize, etc.

**Order :-** Order depicts the order associated with a particular table and the customer.

**Bill :-** Bill is calculated using the order and menu.

**Payment :-** This class is for doing payment. The payment can be done in two ways either cash or card. So payment is the parent class and cash and card are subclasses.

**Cash :-** Payment can be done by cash

**Card :-** Payment can be done by card or online

### Attributes :-

**Hotel Management** :- HotelName, NumberOfEmployees **Employees** :- EmployeeId, EmployeeName, EmployeeSalary **Server** :- ServerId, OrderId

**Chef** :- Chef\_Id, OrderId

**Customer** :- CustomerId, CustomerName, Bill\_Id, OrderId, PaymentId

**Table** :- TableNumber, OccupiedStatus, ServerId, CustomerId

**Menu** :- ItemId, ItemName, Amount

**Order** :- OrderId, ItemId, ItemName, Quantity, CustomerId, ServerId

**Bill** :- Bill\_Id, OrderId, TotalBill

**Payment** :- PaymentId, Bill\_Id

### Methods :-

1. **Hotel Management :-**

**open()** :- This is used to indicate if the hotel is functioning or not.

### Employees :-

**employee details()** :- This method contains the details of the employee.

### Customer :-

**customer\_details()** :- This depicts the details of the customer.

**Ordered items()** :- This method contains the items which are ordered by the customer.

**payment\_status()** :- This says whether the customer paid or not.

### Table :-

**table\_details()** :- This method contains the details of the table along with the customer and no of seats.

**availability\_status()** :- This method says whether the table is occupied or not.

### Menu :-

**items()** :- This method displays the menu items, their availability and their price.

### Order :-

**order\_items()** :- This method orders the items selected by the user from the menu.

### Bill :-

**calculate\_bill()** :- This method calculates the bill for a particular table.

### Payment :

**ispaid()** :- It shows whether payment is successful or not.

### Relationships :

**Inheritance :**

Inheritance is **“is a relationship”**. It has a parent class and its corresponding child classes. The child class inherits the methods and attributes which are required for it from the parent class.

### Association :

In Association, both classes are related to each other but are not physically con- tained with one another. It is known as **“using”** relation. In association relation, consider we have two classes A and B where class A calls class B and Class B also calls class A.

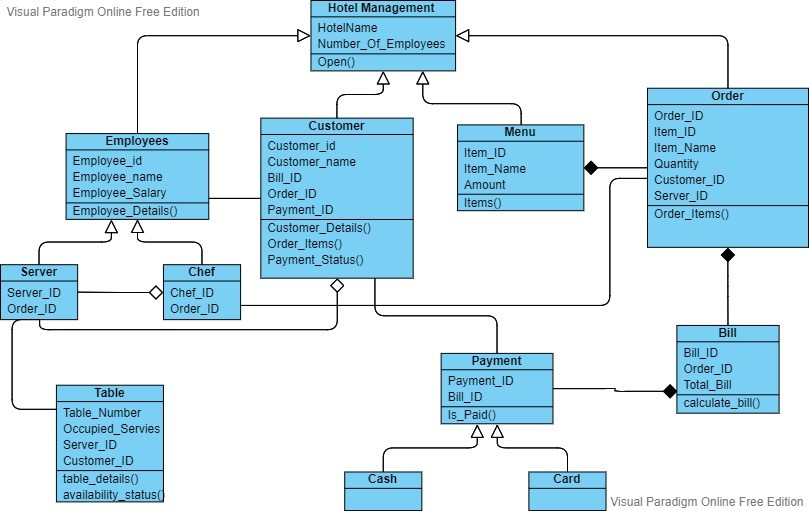
### Composition :

It is also called as **“Has a”** relationship where class A has an instance of class B, Class B is composed inside class A and cannot exist independently without class A. So in composition one class is entirely dependent on another class and is physically contained inside it.

### Aggregation :

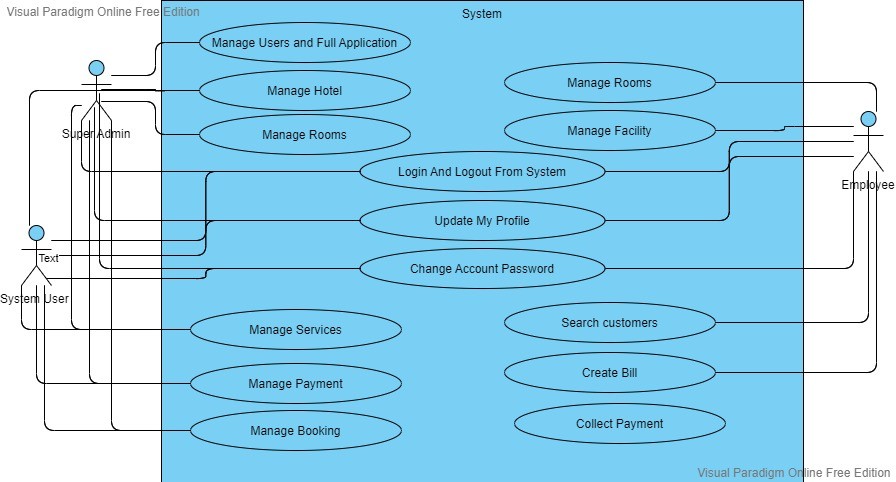
It is also called as **“Has a”** relationship where class A has an instance of class B, but class B is not composed inside class A and can exist independently without class A. So in aggregation, both the classes depend on each other and use each other but none of them are contained inside one another.

Class diagram :-



### 2.) Use case diagram for hotel management system

A **use case diagram** is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.



# Practical :- 4

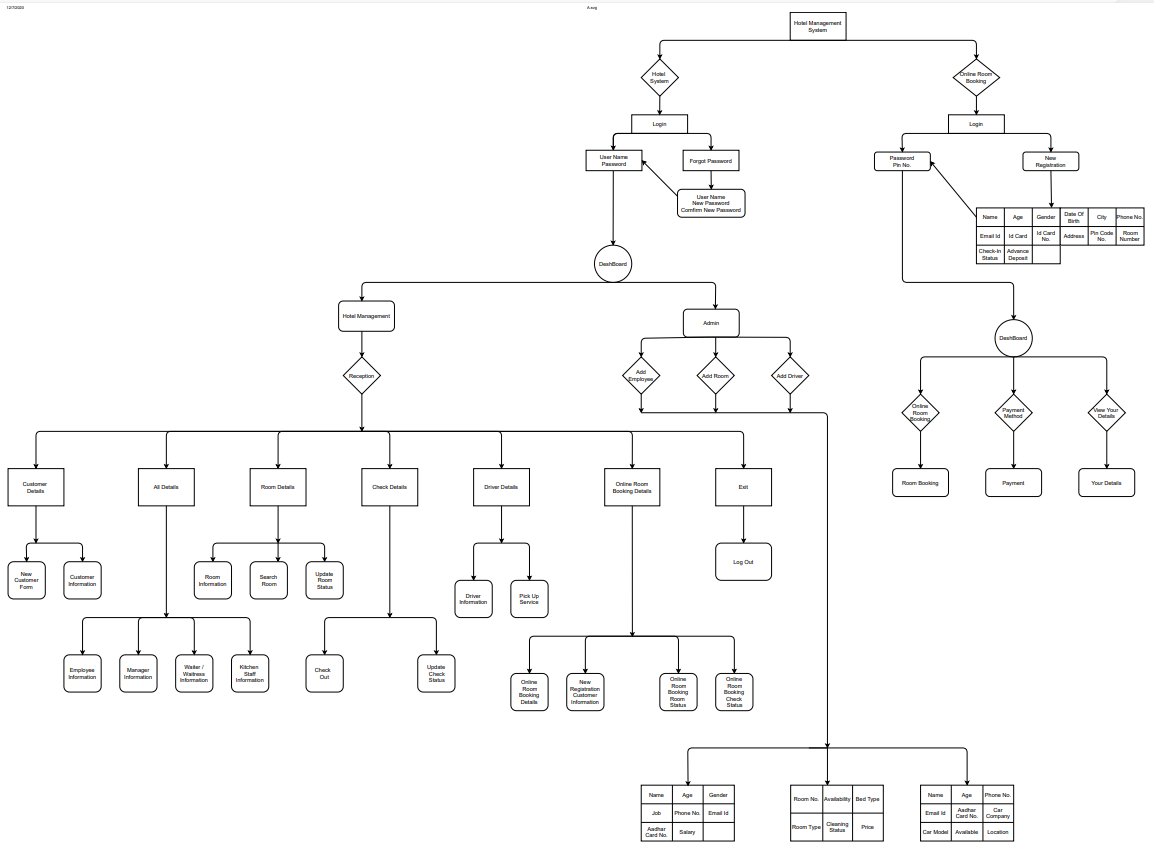
## 1.) Draw ER Diagram :-

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique. The purpose of ER Diagram is to represent the entity framework infrastructure.

### ER Diagram For Hotel Management System

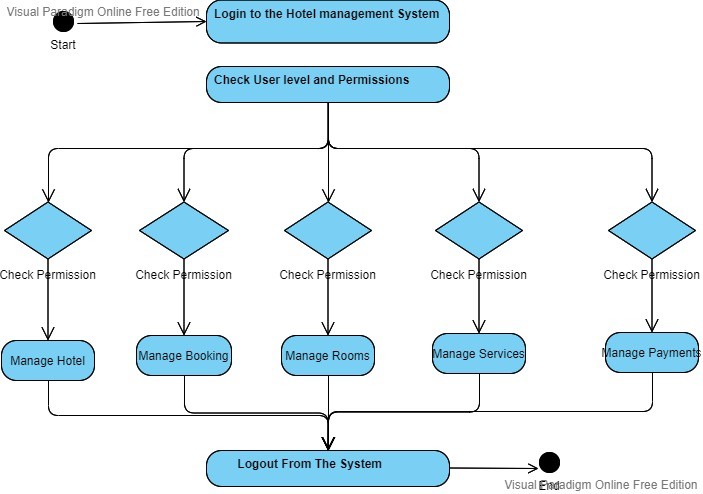


**3.) Draw a activity Diagram**

We use Activity Diagrams to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. We model sequential and con- current activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram.

UML models basically three types of diagrams, namely, structure diagrams, inter- action diagrams, and behavior diagrams. An activity diagram is a behavioral dia- gram i.e., it depicts the behavior of a system.

### Activity DIagram for hotel management system



**Practical 5 :-**

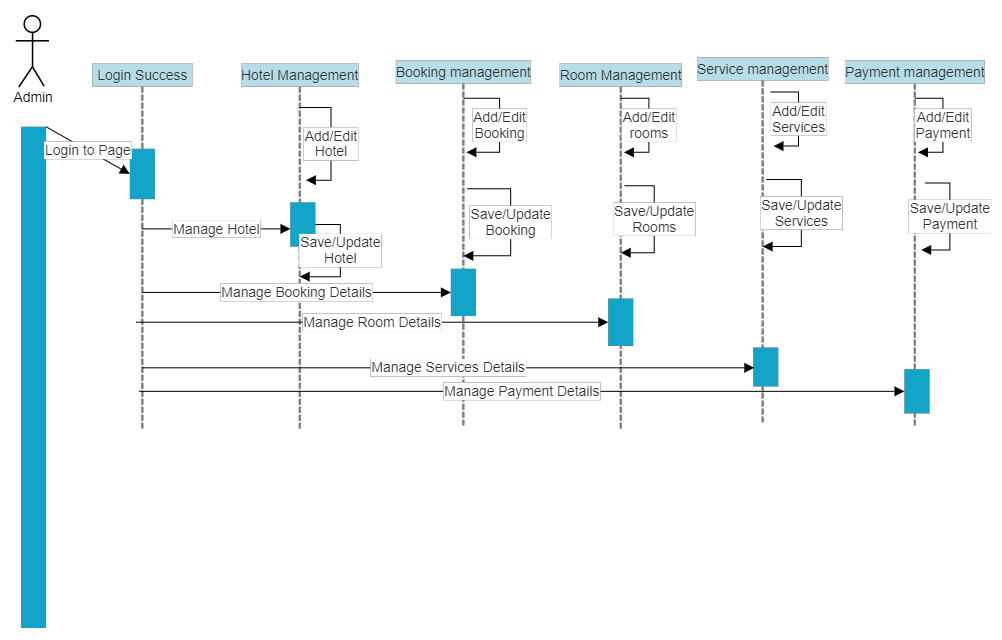
**1.) Sequence Diagram**

A sequence diagram or system sequence diagram (SSD) shows object interactions arranged in time sequence in the field of [software engineering](https://en.wikipedia.org/wiki/Software_engineering). It depicts the ob- jects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of scenario. Sequence diagrams are typically associated with use case realizations in the [logical view](https://en.wikipedia.org/wiki/4%2B1_architectural_view_model) of the system under development. Sequence diagrams are sometimes called event dia-

grams or event scenarios.

\

### Sequence Diagram For Hotel Management System



**2.) Componant Diagram**

Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Now the question is, what are these physical aspects? Physical aspects are the elements such as execu- tables, libraries, files, documents, etc. which reside in a node.

Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.

### Purpose of Component Diagrams

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities. Thus from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc.

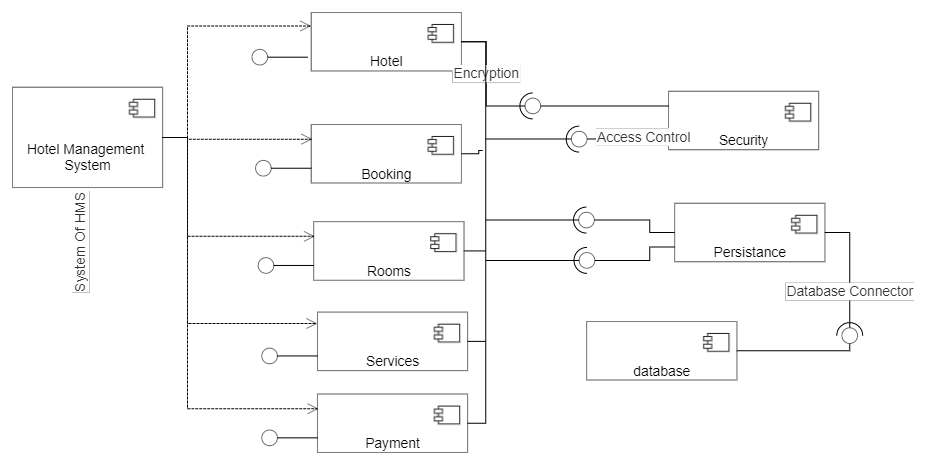
Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

A single component diagram cannot represent the entire system but a collection of diagrams is used to represent the whole.

The purpose of the component diagram can be summarized as − Visualize the components of a system.

Construct executables by using forward and reverse engineering. Describe the organization and relationships of the components.

### Component Diagram for Hotel management system



**practical 6 :- Explain Unit Testing And**

1. **Unit Testing :-**

Unit Testingis a software testing technique by means of which individual units of software i.e. group of computer program modules, usage procedures and operating procedures are tested to determine whether they are suitable for use or not. It is a testing method using which every independent modules are tested to deter- mine if there are any issue by the developer himself. It is correlated with function- al correctness of the independent modules.

Unit Testing is defined as a type of software testing where individual components of a software are tested.

Unit Testing of software product is carried out during the development of an application. An individual component may be either an individual function or a procedure. Unit Testing is typically performed by the developer.

### Objective of Unit Testing :-

The objective of Unit Testing is:

1.) To isolate a section of code.

2.) To verify the correctness of code.

3.) To test every function and procedure.

4.) To fix bug early in development cycle and to save costs.

5.) To help the developers to understand the code base and enable them to make changes quickly.

6.) To help for code reuse.

### 2.) Integration Testing :-

Integration testing is the process of testing the interface between two software units or module. It’s focus on determining the correctness of the interface. The purpose of the integration testing is to expose faults in the interaction between integrated units. Once all the modules have been unit tested, integration testing is performed.

Integration test approaches –

There are four types of integration testing approaches. Those approaches are the following:

### Big-Bang Integration Testing :-

It is the simplest integration testing approach, where all the modules are combining and verifying the functionality after the completion of individual module testing. In simple words, all the modules of the system are simply put together and tested. This approach is practicable only for very small systems. If once an error is found during the integration testing, it is very difficult to localize the error as the error may potentially belong to any of the modules being integrated. So, debugging errors reported during big bang integration testing are very expensive to fix.

### Advantages :-

It is convenient for small systems.

### Disadvantages :-

There will be quite a lot of delay because you would have to wait for all the modules to be integrated.

High risk critical modules are not isolated and tested on priority since all mod- ules are tested at once.

### Bottom-Up Integration Testing :-

In bottom-up testing, each module at lower levels is tested with higher modules until all modules are tested. The primary purpose of this integration testing is, each subsystem is to test the interfaces among various modules making up the subsystem. This integration testing uses test drivers to drive and pass appropriate data to the lower level modules.

### Advantages :-

In bottom-up testing, no stubs are required.

A principle advantage of this integration testing is that several disjoint subsystems can be tested simultaneously.

### Disadvantages :-

Driver modules must be produced.

In this testing, the complexity that occurs when the system is made up of a large number of small subsystem.

### Top-Down Integration Testing :-

Top-down integration testing technique used in order to simulate the behaviour of the lower-level modules that are not yet integrated. In this integration testing, testing takes place from top to bottom. First high-level modules are tested and then low-level modules and finally integrating the low-level modules to a high level to ensure the system is working as intended.

### Advantages :-

Separately debugged module. Few or no drivers needed.

It is more stable and accurate at the aggregate level.

### Disadvantages :-

Needs many Stubs.

Modules at lower level are tested inadequately.

### Mixed Integration Testing :-

A mixed integration testing is also called sandwiched integration testing. A mixed integration testing follows a combination of top down and bottom-up testing approaches. In top-down approach, testing can start only after the top-level module have been coded and unit tested. In bottom-up approach, testing can start only after the bottom level modules are ready. This sandwich or mixed approach over- comes this shortcoming of the top-down and bottom-up approaches. A mixed integration testing is also called sandwiched integration testing.

### Advantages :-

Mixed approach is useful for very large projects having several sub projects. This Sandwich approach overcomes this shortcoming of the top-down and bot- tom-up approaches.

### Disadvantages :-

For mixed integration testing, require very high cost because one part has Top- down approach while another part has bottom-up approach.

This integration testing cannot be used for smaller system with huge interdependence between different modules.

## Practical 7 :- Gantt Chart And Pert Chart

PERT and Gantt charts are visualization tools that are often used in project management. Both of these charts are used for task scheduling, controlling, and ad- ministering the tasks necessary for the completion of a project. The difference be- tween them is that a PERT chart is a kind of network diagram, while a Gantt chart is a bar chart.

### What is the PERT Chart?

PERT Chart is an acronym for (Program Evaluation and Review Technique). A PERT chart is a project management tool used to schedule, organize, and coordinate tasks within a project. It is a method to analyze the tasks involved in completing a given project, especially the time needed to complete each task and to identify the minimum time needed to complete the total project.

### What is a Gantt Chart?

A Gantt chart is a type of horizontal bar chart commonly used in project management, which is a visual view of tasks scheduled overtime. It provides a graphical visualization of a schedule that helps to plan, coordinate, and track specific tasks (or elements) in a project.

Gantt chart boils down multiple tasks and timelines into a single page. Using a Gantt chart allows all stakeholders to perceive the same schedule information, sets mutually understood expectations, and conducts their efforts according to the desired protocol. The Gantt chart tool provides a visual timeline for the start and end of tasks, making it clear how tasks are interrelated and perhaps rely on the completion of another before one can start.

**Practical 2 :- SRS Document**

**Introduction :-**

The regular hotel management system project entirely in an android app. This android application allows the hotel manager to handle all hotel activities in his android phone. Interactive guy and the ability to manage various hotel bookings and rooms from an android phone makes this hotel management system very flexible and convenient. The hotel is a very busy person and does not have the time to sit and manage the entire thing sitting at a single computer. This application gives him the power and flexibility to manage the entire system from a single android phone. Hotel management android project provides room booking, staff management and other necessary hotel management features top make it a complete portable hotel management solution.

**Purpose :-**

The Software Requirements Specification (SRS) will provide a detailed description of the requirements for the Hotel Management System (HMS). This SRS will allow for a complete understanding of what is to be expected from the newly introduced system which is to be constructed. The clear understanding of the system and its’ functionality will allow for the correct software to be developed for the end user and will be used for the development of the future stages of the project. This SRS will provide the foundation for the project. From this SRS, the Hotel Management System can be designed, constructed, and finally tested. This SRS will be used by the system development team which is constructing the HMS and the hotel end users. The Project team will use the SRS to fully understand the expectations of this HMS to construct the appropriate software. The hotel end users will be able to use this SRS as a “test” to see if the constructing team will be constructing the system to their expectations. If it is not to their expectations the end users can specify how it is not to their liking and the team will change the SRS to fit the end users’ needs.

**Document Conventions :-**

The document is prepared using Microsoft Word 2016 and has used the font type 'Times New Roman'. The fixed font size that has been used to type this document is 12pt with 1.5 line spacing. It has used the bold property to set the headings of the document. Use case scenario is written according to Alistair Cockburn’s template. UML diagrams have been created according to UML 2.0 standards. Standard IEEE template is the template used to organize the appearance of the document and its flow.

**Project Scope :-**

The software product to be produced is a Hotel Management System which will automate the major hotel operations. The first subsystem is a Reservation and Booking System to keep track of reservations and room availability. The second subsystem is the Tracking and Selling Food System that charges the current room. The third subsystem is a General Management Services and Automated Tasks System which generates reports to audit all hotel operations and allows modification of subsystem information. These three subsystems’ functionality will be described in detail in section 2-Overall Description. There are two end users for the HMS. The end users are the hotel staff (customer service representative) and hotel managers. Both user types can access the Reservation and Booking System and the Food Tracking and Selling System. The General Management System will be restricted to management users. The Hotel Management System’s objectives is to provide a system to manage a hotel that has increased in size to a total of 100 rooms. Without automation the management of the hotel has become an unwieldy task. The end users’ day-to-day jobs of managing a hotel will be simplified by a considerable amount through the automated system. The system will be able to handle many services to take care of all customers in a quick manner. The system should be user appropriate, easy to use, provide easy recovery of errors and have an overall end user high subjective satisfaction.

**Operating Environment :-**

Hardware: -

1. Operating System Supports all known operating systems, such as Windows, Linux, Mac

2.Computer 512MB+ RAM, monitor with minimum resolution of 1024x768, keyboard, and mouse

3.Hard Drive should be in NTFS filesystem formatted with minimum 10 GB of free space.

Software: -

1.Software is designed to run on any platform above Microsoft Windows 7 (32bit). 2.Microsoft .NET Frameworks 4.0 or above.

3.Microsoft SQL Server Management Studio Express 2010.

**Design and Implementation Constraints :-**

Software development crew provides their best effort in developing the system. In order to maintain the reliability and durability of system, some design and implementation constraints are applied. Availability of an android app for hotel management system could make the system portable but due to time constraint it is not possible. System will need a +minimum memory of 512MB. But it is recommended to have a memory of 1GB. When designing interfaces of system, we had the capability of work with new tools such as Dev Express. Considering the client’s budget, we decided to create those interfaces in a simple realistic manner using affordable technology.

**Assumptions and Dependencies :-**

Some software used in implementing the system is with high cost and the client has agreed to afford the amount of money needed to purchase them. It’s assumed that client won’t change that decision on the next phases of the software development. Although we assume that client is using windows 7 or windows 8. Otherwise if client use an open source operating system, there is a need of changing the SRS accordingly