Online Hotel Management System

* Project O.H.M.S is web based application, which allows Hotel manager & owner to handle all the operations by sitting anywhere
* The purpose of this project is to provide the information of hotel, online room booking facilities & automatic bill generation
* This s/s is very easy and attractive & make the customers comfort
* By using this s/s we can overcome below problem
* it makes easy for admin to track the records of booking, it saves lot of time.
* Several operational issues like managing reservations, performing desk operations, ~~maintaining~~ ~~cleanliness~~ in every room & hotel premises. & many more hotel staff fails to maintain.

Features of O.H.M.S:-

* E payment process
* Reservation Emails
* Mobile & tablet optimized
* Storing of guest profiles.

future Scope:-

* Ca if Customer cancelled the room, that room will be given to the other Customer who needed
* Adding of Worker module so that Admin can intimate Worker to clean the room, once the visitor vacate.
* Email generation task is to be integrated.
* Onsite-quiries should be answered.

* Scope of the project:-

* Helps in preplanning of the customer on Booking the room in desired hotel if there is
* Availability.
* Increase your online visibility.
* prevents Double Booking and manual errors
* Accurate daily reports

Limitations:- * Reduce the Connection (b/w staffs & Human touch guests)
* Relay to internet connection
* Risk of Cyber-Attack.

Common function of managed bean

* Validating a component's data
* ~~Pa~~ for navigation purpose.

Session scope:- it can persist across multiple HTTP requests in a Web application.

* Transaction interface defines unit of work
* It maintains abstraction from implementation of transaction
* It is associated with session & instantiated by calling session.beginTransaction()
* Save() is used for inserting record
* t.commit():- ~~Saving~~ saves all the changes done
* SQL date represents date without time
* Util date represents both Date & time

7

1 day = 24hrs
1hr - 60min
1m - 60sec
1se = 1000m:S

11 days - 25 times

<h:inputText> - to render an i/p field on web page

<h:secretText> -

* JSF is a server side java framework for web development

*

JSF life cycle divided into two main phases

1. Execute phase
2. Render phase

} JSF application begins when client makes an HTTP req for a page & ends when server responds with the page

benefits of JSF :-

* clean & clear separation b/w behaviour & Presentation of web application
* We can write business logic & user interface separately

features of JSF

* Components - Based fw
* Implements Facelets Technology
* Integration with Expression language
* Supports HTML5
* Bean annotations

* for JSF in a computer, platform needs to have JDK 1.5 (8) above

MVC - Model View Controller

Java, Hibernate, JSP, JSTL

Hibernate :- It is an ORM (Object Relational Mapping (Persistence) and Query service for any java application.

* Hibernate maps java classes to database tables.

* Difference B/w String and StringBuilder

String	StringBuilder
* String objects are immutable	* String Builder objects are mutable

String Builder	String Buffer
* String Builder objects are not synchronized & not thread safe	* String Buffer objects are synchronized & thread safe

* OOPL? → Any programming lang which supports class, object, Encapsulation, Abstraction, inheritance, polymorphism

* class :- class is a blueprint in which objects are created

* Object :- it is a real world entity, we can identify state, behavior, identity

Keyword for creating object, we use (New)

* Constructor :- It is used to initialize an object

* Constructor name should be same as class name

* It has no return type.

What is default Constructor:- if a programmer does not supply any constructor for a class then java compiler supplies a default constructor for the class

Interface:- Interface contains methods without body interface can also contain final variables

* We can't create object to the interface

* Implements keyword.

Encapsulation:- Binding the data into a single unit.

Abstraction:- hiding unnecessary details and showing only necessary data

Inheritance:- acquiring properties from one class to another class.

* reduce code duplication.

Polymorphism:- Single action can be done in different ways

① C.S :- compile time / static polymorphism :-
Ex:- method overloading

② R.D :- Runtime / Dynamic polymorphism :-
Ex:- method overriding

over loading:- 2 or more methods having same name and different parameters in given class

```
Public class B
{
  Public string M1 (int x)
    {
    return ("Hi")
    }
  Public string M1(int x, int y)
    {
    return (Hello)
    }
}
```

overriding:- sub class has same method as decla. in the parent class, it is known as o.r

* subclass provides a particular implementation of methods declared by one of its parent class

Session factory:- S.F in 'H' is responsible for creation of session objects

* It is thread safe
* It is a heavy weight object. bcz, d.B, mapping, config file → it maintains. so it is to create heavy weight object and kept for latemre
* S.F objects are one per application

Session:- * session objects are 1 per client
* it provides crud interface for mapped class
* not thread safe.
* easy access to API

Entity class :- To store point objects in database

* It is a POJO class (plain old java objects)
* It reduces development time & allows easy mapping of business objects.

class	entity
1. Syntactic	1. Semantic (logical)
set of things having same property (or) attribute in common & are differentiated by kind, quality etc	collection of fields associated with d.B

Getters and Setters :- These are used to protect the data especially while creating the classes.

* getters are used to return its value.
* setters are used to set (or) update the value

Param :- Param maps requested parameter name to single value.

* Criteria :- it is like where clause
* req. get parameter :- it is criteria language in which we can perform the required operation
* Projection :- projection is used for particular field.

open session:- This method helps to open new session if you are done with data base operations you can close the session.

Key Components of Hibernate Configuration Object:-

* D.B Connection — Handled by 1 (or) more config files
* class mapping setup (connection b/w java class & D.B table)

Collections In Hibernate:-

* A collection basically a List, Set, Map, Sorted set, sorted Map.
* Hibernate injects persistance collection based on type of interface

collection types in Hibernate:-

ABSLM

Array, Bag, Set, List, Map

Persistance Table:- It is used with entity beans to define the corresponding table name in D.B

I.D:- defines primary key in entity bean.

Embedded ID:- Defines Composite key in entity bean.

Column:- helps in defining Column name in D.B table.

Generated value:- It is used for generation of Primary key

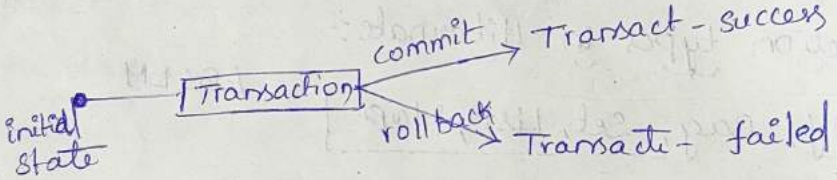
Get current Session:- it creates new session, if it doesnot exists (or) uses current hibernate context

* it automatically close the session once the transaction done.

Benifits of Hibernate:-

* Automated session closing abilities
* Exception Handling is automated.

first level cache	Second Level cache
it is maintained at session level	it is maintained at session factory level



Thread Safe:- code will work even if many threads are executing it simultaneously.

Save():- it is used to store objects in d-B

* Save() only for insert records
* Save or update() can either insert (or) update methods

Session.get():- It always hit the database and returns real objects

Session.load():- it always returns a proxy without hitting D-B) proxy is an object with the given identifier value, properties are not instantiated yet, it just look like temperory take objects

1. What is abstract class?

Abstract class is a class which contains zero or more abstract functions

* We can't create the obj for abstract class

```
abstract class test
{
  Public void fun1()
  {
    S.O.P.ln("Welcome to Bglr");
  }
  public abstract void fun2();
}
```

* it does not support multiple inheritence
* it has final, non-final, static & non-static Variable
* Abstract keyword is used to declare abstract class
* It can be extended by extends keyword

Interface:- A method without body and it contains only signatures - it contains only final variable & static variable.

* it has only abstract methods.
* Support multiple Inheritance
* ~~inter~~ implements keyword is used
*

We can't create object to interface

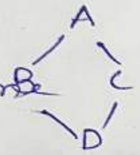
Ex:
```
Public interface MyInterface
{
    Public void fun1();
}
Public class implements MyInterface
{
    Public void fun1()
    {
        // implement body
    }
}
```

Diff b/w interface & abstract class

interface	Abstract class
doesn't contain Data Member	contains D.M
" " Construct	" Constructor
Contains only incomplete member	both incomp complete me
Can't have access modifier by default everything is assumed as public	contains access modif for subs, functions properties
members can not be static	Complete mem of Abst class only stati

4. What is diamond problem & explain with ex?

The diamond problem is an ambiguity that arises when two classes B and C inherit from A, and class 'D' inherit from both B & C. ~~If method~~ in 'D' calls a method defined in 'A' (& doesn't override the method) ~~B & C have overridden that method differently~~, We need not to override these methods, BCZ java doesnot support multiple inheritance



21. 2 2
22. compile ~~time~~
23.
25. compile time

26.
27. compile
28. welcome

~~Example~~ :- 30. 123

company
return type String ← toString, equal, getClass, hashCode

Madhu a-18

10 +

Example :-

```
class B {
    p. v. foo() {...}
}.
class C {
    p. v. foo(){..}
}
class D extends B. C {
}
new D(). foo();
```

(10). Arg list
C. Madhu
11.
12. compile time
13. compile
14.
16. protected
18. false
19.
20 hibye

31. compile
32. Hi
33. Common 'A' and 'D'
34. compile 36. Hi
35 True

Online Hotel Management System

* project online H.M.S is a web based application that allows the H.Manager & owner to handle all hotel activities online easily & safely.

* (like import hibernate from)
* We import criteria from org.hibernate.Criteria

* Session:- provides crud interface for mapped class (NTS)
* Session factory :- responsible for creation of session objects

Customer ~~login~~ Register :- Before login to customer page, * Customer need to register, once the registration done then the customer should login after the customer login then the list of Availability of hotels will be appear, then the customer should select the ~~hotel~~ desired hotel. and search for the rooms which are available. then he/~~she~~ customer Book the room. based on their need.

——

* JDK - 1.8 Version
* JSTL - 1.2
* mysql - 8.0.30
* Hibernate - 4.2.5.0 (5)
* Html - 5
* css - 5
* eclipse - 4.25.0

4 - layers

* Java application
↓
Persistent object
↓
Hibernate.properties & XML Mapping
↓
D.B.

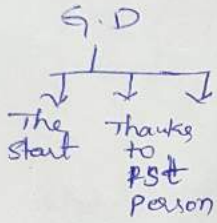
Why do we use hibernate?

* 'H' makes the application development process easy.
* 'H' allows ~~the~~ to map the java objects to relational D.B
* Saves development time

Advantages of EJB:

* EJB have remote access capabilities which enables clustering of data
* EJB simplifies the development of small & large enterprise applications.

S.D



According to current trending technology.

* life without mobile phone is much difficult
* Now a days mobile has played significant role in peoples life.
*

---

* it allows us to share the important msgs with our family
* with the help of mobile phone we can transfer money with in seconds.
* it saves lot of time using internet
* with the help of mobile we can connect globally.
* We can come to know what is happening in the world. also it is used for reducing the stress, we can watch