## PROJECT REPORT ON “MAKE MY TOUR”

**Project Report Submitted By Miss/Mr.**

## 4th Semester MCA BPUT Reg. No.-1705280018



**Under The Guidance of Prof. Pravakar Mishra**

## HOD, Department of Master of Computer Application

Project Report submitted to Institute of Management and Information Technology(IMIT) in partial fulfilment of the requirements of 4th Semester MCA examinations of BPUT, ODISHA-**2019**



## Institute of Management and Information Technology(IMIT), Odisha Pin Code-752054 www.imit.ac.in

**CERTIFICATE**

This is to certify that the project work entitled **MAKE MY TOUR** done by **Miss/Mr…………** bearing

**Regd.No………………………..** is an authentic and original work.

Signature Signature

(Internal Examiner) (ExternalExaminer)

Date.................. Date.................

Signature of the Student: Signature of the Guide

Date:………………….. Date :………........

Name: Designation:HOD

INCLUDE COMPANY CERTIFICATE

**DECLARATION**

I Bhanupriya Nayak , Regd. No.:1705280018 do hereby declare that the project report entitled **MAKE MY TOUR** submitted to Institute of Management and Information Technology(IMIT) **BPUT University,**

**Rourkela** for the award of the degree of **MASTER OF COMPUTER APPLICATION (MCA) ,** is an authentic and original work carried out by me from 15th Dec ,2022 to 25th March,2022 at LCC under the guidance of **Mr. Bibhu Ranjan Mohanty** and **Prof. ……………….**

Signature of the student Date:30th june 2022

**ACKNOWLEDGEMENT**

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people whose constant guidance and encouragement crowns all efforts with success. The acknowledgement would like to express deep gratitude respect to all those people behind the screen who guided inspired and helped me for the completion of my project work.

I would like to express my deep sense of gratitude towards my guided Teacher **Prof. …………….** HOD of MCA Dept. for their sincere guidance, valuable suggestion and constant encouragement during the preparation of the projects report.

**Your Namw** Date: 30th June, 2022 4th Semester, MCA Regd.No:

|  |  |  |
| --- | --- | --- |
| **S.No.** | **CONTENTS** | **PAGE No.** |
| 1 | Abstract | I |
| 2 | List of Figures | II |
| 3 | List of Tables | III |
| 4 | **Chapter-1** |  |
|  | 1.1 Introduction |  |
|  | 1.2 Problem Statement |  |
|  | 1.3 Proposed System |  |
|  | 1.4 Expected Benefits |  |
|  | 1.5 Objective |  |
|  | 1.6 Scope |  |
| 5 | **Chapter-2** |  |
|  | **2.1 System Analysis** |  |
|  | 2.1.1 Background Research |  |
|  | 2.1.2 Requirement Analysis |  |
|  | 2.1.3 Make My Tour System Requirements |  |
|  | 2.1.4Feasibility Study |  |
|  | 2.1.5 Initial Project Schedule (PERT CHART) |  |
|  | * + 1. System Requirements        1. User Requirements 2.1.6.2Functional Requirements        2. Non Functional Requirements        3. Hardware and Software Requirements |  |
|  | **2.2 System Design** |  |
|  | 2.2.1Software Engineering Paradigm applied |  |
|  | 2.2.2 Modularization details |  |
|  | 2.2.3 Process Flow Diagram |  |
|  | 2.2.4Analysis Model |  |
|  | 2.2.5Data Flow Diagram |  |
|  | 2.2.6 Use Case Diagram |  |
|  | 2.2.7 Activity Diagram |  |
|  | 2.2.8 Class Diagram |  |
|  | 2.2.9 State Transition Diagram |  |
|  | 2.2.10 Sequence Diagram |  |
|  | 2.2.11 Collaboration Diagram |  |
|  | 2.2.12 Database Design   * + 1. ER Diagram     2. Data Tables |  |
|  | 2.2.13User Interface Design |  |
| 6 | **Chapter-3** |  |

|  |  |  |
| --- | --- | --- |
|  | **3.1 Tools And Technology Used**   * 1. Development Tools   2. Technology Used   3. MySQL |  |
| 7 | **Chapter-4** |  |
|  | **4.1 Coding** |  |
|  | 4.1.1 Comments and Description |  |
|  | 4.1.2 Standardization of the coding /Code Efficiency |  |
|  | 4.1.3 Error handling |  |
| 8 | **Chapter-5** |  |
|  | **5.1. Testing** |  |
|  | 5.1 .1 Testing techniques and Testing strategies  5.1.2 Test case designs |  |
| 9 | **Chapter-6** |  |
|  | **6.1 System Security measures** |  |
| 10 | **Chapter-7** |  |
|  | **7.1 Screens** |  |
| 11 | **Chapter-8** |  |
|  | * 1. Future scope and further enhancement of the Project   2. Conclusion |  |
| 12 | **Chapter-9** |  |
|  | 9.1 Bibliography / references |  |

1. **Abstract:**

This project is aimed at developing web based **MAKE MY TOUR** for customers who wants to travels tourism place worldwide. This MAKE MY TOUR project can be used to store the details of the customers, update the packages based on the customer demand, produce money receipts for customers, generate different reports etc. This is one integrated system that contains both the user component (used by customers, advertise company etc) and the admin component (used by the administrators for performing admin level functions such as adding new category to the tour, adding new subcategory, adding new package, changing the price of package, produce report etc.)This system runs on multiple terminals, offers a GUI interface to its users and connects to a common database(s).Since it is a web based application the Administrators or any high level authority can track details about the customer’s booking reports , monthly expenses etc. In addition to this admin can generate various reports on the basis of daily, monthly or yearly, which acts as important instruments for taking proper decision towards the benefit of tour business and to meet current tour business challenges. This system provides beautiful and clear interfaces for easy use of system.

## List of Figures:

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Figure Name** | **Page No** |
| Figure 1 | Feasibility Evaluation Grid Showing Comparison of System Alternatives | 7 |
| Figure 2 | Problem Solving Loop | 12 |
| Figure 3 | Water Fall Model (SDLC) | 13 |
| Figure 4 | Data Flow Diagram (Level 0) | 16 |
| Figure 5 | Data flow diagram (Level 1) | 17 |
| Figure 6 | Use Case | 18 |
| Figure 7 | Activity Diagram for report generation | 18 |
| Figure 9 | State Transition Diagram For Sales | 19 |
| Figure 10 | Sequence Diagram | 19 |
| Figure 11 | Collaboration Diagram | 20 |
| Figure 12 | E-R Diagram | 21 |
| Figure 13 | Eclipse IDE. | 26 |
| Figure 14 | Apache Tomcat/9.0.1 | 28 |
| Figure 16 | JDBC API for 2 & 3 Tier | 33 |
| Figure 17 | Testing Types | 61 |

1. **List of Table:**

|  |  |  |
| --- | --- | --- |
| **Table No** | **Table Name** | **Page No** |
| Table 1 | Project Schedule | 8 |
| Table 2 | Functional Requirement | 9 |
| Table 3 | Hardware Requirement | 11 |
| Table 4 | Software Requirement | 11 |
| Table 5 | tbl\_category Table | 22 |
| Table 6 | tbl\_ subcategory Table | 22 |
| Table 7 | tbl\_ enquiry Table | 22 |
| Table 8 | tbl\_package Table | 23 |
| Table 9 | tbl\_usersTable | 23 |
| Table 10 | Category View page change test case | 63 |
| Table11 | Package View page change test case | 63 |
| Table 12 | Subcategory View page change test case | 64 |
| Table 13 | User view page change test case | 64 |
| Table 15 | Report Generation page change test case | 65 |
| Table 16 | Login authorization test case | 65 |
| Table 17 | MySQL / Apache Tomcat Server upgrade | 66 |

## CHAPTER-1

**INTRODUCTION**

* 1. **Introduction:**

In software industry requirement engineering is one of the most important parts of software engineering process, which one gives us the proper scenarios what the customers want, analyzing their needs and checking the feasibility what they need, negotiating a reasonable solution etc. In software industries, a software project begins when a business need is identified. So the first step is we need to understand the customer needs. Figure out a rough feasibility analysis, not only the customer’s need but also with the people who are apparently involved with the introducing system. In this phase we get some requirement for online inventory system.

**Make My Tour** is fully integrated, which means users are able to work more efficiently and management decision making. Software Solutions is an IT solution provider for a dynamic environment where business and technology and end users strategies converge. Their approach focuses on new ways of business combining tour innovation and adoption while also leveraging an organization’s current manual tour.

## Problem Statement:

1. The customers/visitors need to ask information and review immediately to make sure that what place they should go.
2. Application or website that include with travel agency such as trip advisor usually use simple Forum.
3. The information of maps that have download to use with no internet connection may be wrong and not enough for customers demand.
4. About how to go with different transportation, ways to go, price that have to manage, and who that customers go with.
5. The search with stronger type such as when customers/visitour want to with family and their pets, what place that they should go.
6. With history of systems maybe tell customers/visitors just place that they go.
7. In nowadays, with many social network such as Facebook and line have function about travel but they doesn't have function agency inside.
8. Different place different language, If foreigner ask about information and how to go in the local area what should customers do.
9. Travel agency in nowadays is usually local system, not fast such as when customers booking flight or hotel, agency will give a ticket and customers must bring the ticket to counter airline.

## General Requirements:

Create the application or website with **social network** and co-operate with another company such as Facebook, grab taxi (transportation), bank(SCB), google(map, drive).Make stronger **information of how to travel** and everything about **maps** and **search**. **Search** with more type of people such as go with family, couple or single , tour with friends , and with their pets and type of place such as hotels, flights, restaurants, hospital, toilets, theme park, beaches, etc. The customers can **add** place that doesn't have in search and map using fill in website helper. Systems that useful with tour.User interface with more interesting. **Booking** using application by email and barcode. The customers can **review**

**\*** any place and see what place that people or their friends have ever go to with rates, comments, photos, videos, seasons, and promotions, information of that place , status such as traffic, repairs, shortcut.

## Proposed Solution

To debug the existing system, remove procedures those cause data redundancy, make navigational sequence proper. To provide information about users on different level and also to reflect the current work status depending on customers/visitors.

## NEED FOR COMPUTERIZATION

We all know the importance of computerization. The world is moving ahead at lightning speed and everyone is running short of time. One always wants to get the information and perform a task he/she/they desire(s) within a short period of time and too with amount of efficiency and accuracy. The application areas for the computerization have been selected on the basis of following factors:

* + - Minimizing the manual records kept at different locations.
    - There will be more data integrity.
    - Facilitating desired information display, very quickly, by retrieving information from users.
    - Facilitating various statistical information which helps in decision-making?
    - To reduce manual efforts in activities that involved repetitive work. Updating and deletion of such a huge amount of data will become easier.

This software system must be portable and platform independent (if possible). In light of this, the

recommended software development tools include JSP on the front-end, and MySQL as the back-end DBMS.

* 1. **Expected Benefits**

The Make My Tour will bring a number of benefits to the customers/visitors:

* + The system will help to keep accurate records of tour for visitors.
  + The maintenance of accurate records should reduce misappropriation.
    - The customers can view or **find the information** of tourist attraction.
    - The customers can **search for booking** tour place time to go in advance.
    - The customers can find the information and the **calculate system** of time, ways, price to the destination place.
    - The customers can **review** place or find another friends and people reviews.
    - The customers can see **private history** with analyse with booking, transport, price, time, etc. And keep or share the information.
    - The customers can use function **dictionary** to translate sentences for communication with different language of travellers.
    - The customers can **note** with text or plan trip with themselves.
    - The customers can find the **location** themselves and share location if they want.
    - The customers can see **whether** of each area inside map or place that they want to go with Google map detection.
    - The customers can view different package photo using **website**.



## Objective:

The Make My Tour will fulfill the following objectives:

* + - Features to facilitate entry and update of package, category, subcategory and view enquiry details.
    - Tracking of customers/visitors contact information
    - Tracking of booking details of customers/visitors.
    - The customers can view or **find the information** of tourist attraction.
    - The customers can **search for booking** tour place time to go in advance.
    - Facility to generate an invoice by administrator.
    - Alerts for the booking reorder of customer

## Scope:

The main components of the system will be as follows:

* + - A relational database in MySQL or another portable DBMS stores the essential data for subsequent analysis
    - An Acquisitions Subsystem (AS) facilitates tracking of purchases, utilization and sale of inventory items
    - A Financial Management Subsystem (FMS) provides the financial implications of various transactions
    - Point of Sale Subsystem (POSS) manages interfacing between the point of sale equipment (barcode scanners, and invoice printers, etc.) and the internal database

## CHAPTER-2

**SYSTEM ANALYSIS & DESIGN**

## Background Research

We started research by identifying the need of **Make My Tour** for visitors who facing problem for booking tour. Initially we bounded our research to find the general reasons that emerged the needs of Make My Tour. We used different techniques to collect the data that can clearly give us the overall image of the Website. The techniques we used were interview with the developers, visiting online websites that are presented as the templates and visiting some visitor to see their MMT application. Basically the following factors forced us to develop Make My Tour application:

*Cost and affordability

**Make navigational sequence proper. Effective flow of tour management.

* Difficulty in monitoring the inquiry and booking history worldwide.

*Making smooth and effective UI design in website.

*Reduce the paper work and moving towards digital environment

## Requirement Analysis

We collected a number of requirements for project from our primitive research, website visits, and interview to the concerned personnel and their experiences regarding the concepts of its development. We have even visited some visitors in Bhubaneswar and other part of state and analyze its importance and try to develop the project by fulfilling all the weakness that were found in the application. We then decided to build same type of website with different logic flow and new language which will be suitable for the website.

## Make My Tour System Requirements:

The goal for the application is to manage the Make My Tour function of the a agency. Once it is automated all the functions can be effectively managed and the organization can achieve the competitive advantage.

**Business requirement** are discussed in the Scope section, with the following additional details:

* + - * User Friendliness is provided in the application with various controls provided by system Rich User Interface.
      * The system makes the overall project management much easier and flexible.
      * It can be accessed over the Intranet.
      * The employee information can be stored in centralized database which can be maintained by the system.
      * This can give the good security for user information because data is not in client machine.
      * Authentication is provided for this application only registered Admin and every visitors can access website.
      * The automated system will provide to the visitors for reliable services.
      * The speed and accuracy of this system will improve more and more.

**Overview of Storage Requirements**

It is anticipated that the system will contain the following main information entities:

* Users Management
* Tour Category Management
* Tour Subcategory Management
* Package Management
* View Inquiry
* Booked Invoices Generation
* Payments Received
* Financial Transactions
* Ad Management

**Anticipated Outputs:** It will be possible to run queries and obtain reports on all information stored. As such, operations will be provided to extract and display information from the above mentioned information entities. Some of the more prominent outputs are as follows:

* + Users Listings
  + Tour category Listing
  + Tour subcategory Listing
  + Package Listing
  + Inquiry Listing
  + Invoice History
  + Payment History
  + Purchase Invoices
  + Account Balances — Accounts Payable and Accounts Receivable
  + Financial Transactions Log
  + Investments Log

## Feasibility Study

Preliminary investigation examines project feasibility; the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All systems are feasible if they are given unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

Technical Feasibility Operation Feasibility Economical Feasibility

* + - 1. Technical Feasibility:

The technical issue usually raised during the feasibility stage of the investigation includes the following:

* + - * 1. Does the necessary technology exist to do what is suggested?
        2. Do the proposed equipments have the technical capacity to hold the data required to use the new system?
        3. Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
        4. Can the system be upgraded if developed?
        5. Are there technical guarantees of accuracy, reliability, ease of access and data security?
    1. Operation Feasibility:

Proposed projects are beneficial only if they can be turned out into information systems, which will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation.

Some of the important issues raised are to test the operational feasibility of a project includes the following: –

* + - 1. Is there sufficient support for the management from the users? Will the system be used and work properly if it is being developed and implemented?
      2. Will there be any resistance from the user that will undermine the possible application benefits?
      3. This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

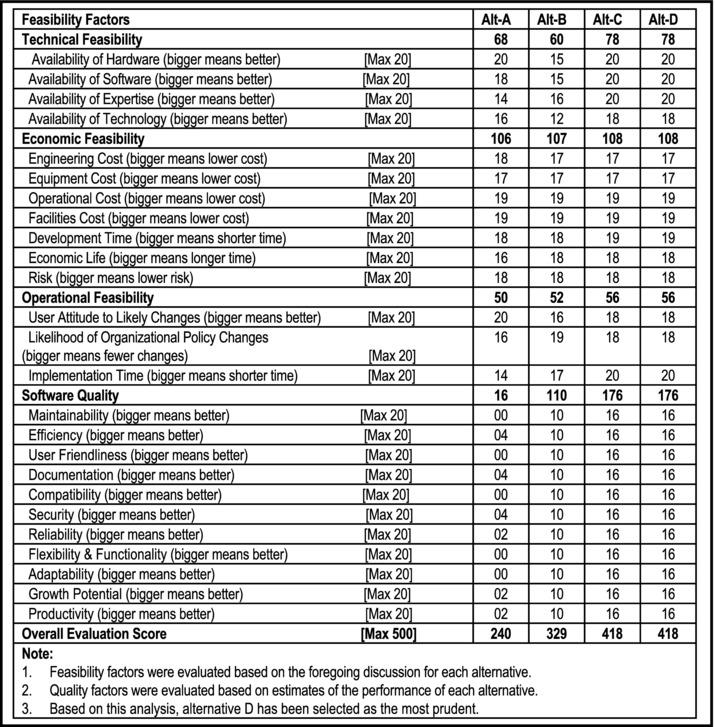
2.1.4.3 Economical Feasibility:

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial

benefits must equal or exceed the costs. The system is economically feasible. It does not require any additional hardware or software.

## Evaluation of Alternatives

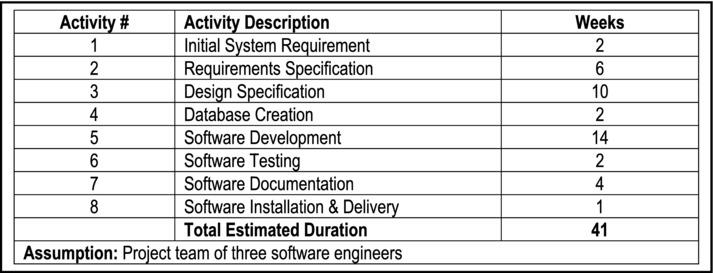
The following table provides a comparison of the project alternatives, based on a number of feasibility factors. For each factor, a rank in the range {0 ... 20} is given for each alternative.



*Figure. Feasibility Evaluation Grid Showing Comparison of System Alternatives*

## 2.1.5 Initial Project Schedule

Following Table shows an initial project schedule. This will be further refined as more details become available. The estimated duration (assuming three full-time software engineers) is 41 weeks.



## System Requirement

Requirement Analysis presents the requirement specification, software and hardware requirements for both system developers and system users, process model and data model.

In this section we specify the External Interface Requirements, functional and nonfunctional requirements of the system.

## User Requirements

The main users in this system are administrators and visitors. By using this system, users can access Make My Tour with some function. The system will allow administrator to login and maintain requirement in the update package details, category details, and subcategory details and view inquiry of visitors. Normal visitor are allowed sent inquiry of suitable package from available of different package. Make Tour administrator can check a receipt and print out the bill as reference of package booked.

## Functional Requirements

A requirement specifies a function that a system or component must be able to allow the user to perform some kind of function. The main purpose of the system is to increase visitors booking by creating the complete add package records, add category and subcategory etc.

|  |  |  |  |
| --- | --- | --- | --- |
| No | Requirement name | Requirement Description | Priority |
| 1 | Log in | This function will enable  administrators to log in to the system. | Mandatory requirement |
| 2 | Change password | This function will enable administrators to change  password. | Mandatory requirement |
| 3 | Update password | This function will enable administrators to change  password. | Mandatory requirement |
| 4 | Add Category | This function will enable the  admin to add category. | Mandatory requirement |
| 5 | Update Category | This function will enable the  admin to update category. | Mandatory requirement |
| 6 | Delete Category | This function will enable the  admin to delete category. | Mandatory requirement |
| 7 | Add Subcategory | This function will enable the  admin to add subcategory. | Mandatory requirement |
| 8 | Update Subcategory | This function will enable the  admin to update subcategory. | Mandatory requirement |
| 9 | Delete Subcategory | This function will enable the  admin to delete subcategory. | Mandatory requirement |
| 10 | Add Package | This function will enable the  admin to add package. | Mandatory requirement |
| 11 | Update Package | This function will enable the  admin to update package. | Mandatory requirement |
| 12 | Delete Package | This function will enable the  admin to delete package. | Mandatory requirement |

|  |  |  |  |
| --- | --- | --- | --- |
| 12 | View reports | This function will enable the inventory managers to view  reports. | Mandatory requirement |
| 13 | Generate reports | This function will enable the inventory managers to  generate reports. | Mandatory requirement |
| 14 | Check exception | This function will enable the sales manager to check  exception | Mandatory requirement |
| 15 | Check receipt | This function will enable the sales manager to check  Receipt | Mandatory requirement |
| 16 | Check Inquiry | This function will enable the admin to check inquiry of  visitors. | Mandatory requirement |
| 17 | Confirm Inquiry | This function will enable admin  to check return item | Mandatory requirement |
| 18 | Category View | This function will enable the  admin and visitors. | Not mandatory  requirement |
| 19 | Subcategory View | This function will enable the  admin and visitors. | Not mandatory  requirement |
| 20 | Package View | This function will enable the  admin and visitors. | Not mandatory  requirement |
| 21 | Send Inquiry | This function will enable to  visitors. | Not mandatory  requirement |

## Non Functional Requirements

* The Administrator must be able to access their account 24 hours a day, seven days a week.
* The non-functional requirements of the adminstrator systems are as described below.

## Security

* + The system must have protection from unauthorized users. By using username and password.
  + Also system show error if the username or password is not correct
  + To use the system, every user has to login by key in their username and password.

## Performance

* The time system required to process and respond when a user key in the Identification that is username and password
* The system is supposed to respond in less than 2 seconds.

## Availability

* Make My Tour system is a standalone system which has four to five (4 – 5) users and users must have to use it during operating hours

## *Hardware and Software Requirements Hardware Requirement

|  |  |
| --- | --- |
| Hardware | Minimum Requirement |
| Processor | Core 2 i3 or above |
| RAM | 2 GB |
| Hard-disk Space | 120 GB |

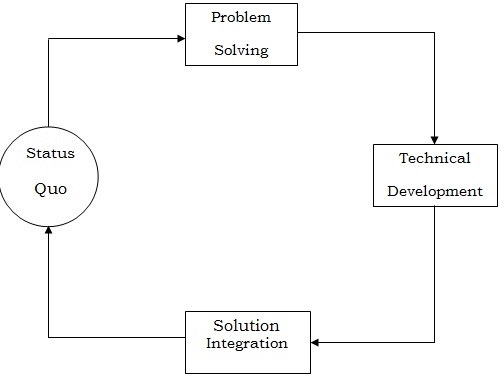
***Software Requirement**

|  |  |
| --- | --- |
| Software | Requirement |
| Operating System | Window 7 or more |
| Tools | MySQL |
| Technology | JSP,Servlet,HTML,CSS |
| JavaScript | Scripting Language |
| Documentation Tools | Microsoft office 2007/10 |

## System Design

* + - * 1. **Software Engineering Paradigm applied (Process Models)**

The development strategy that encompasses the process, methods, and tools and the generic phases is called Software Engineering Paradigm. The s/w paradigm for software is chosen based on the nature of the project and application, the method and tools to be used, and the controls and deliverables that are required. All software development can be characterized as a problem-solving loops (fig. 2) in which four distinct stages are encountered: status quo, problem definition, technical development, and solution integration.



The four basic models are:

* Waterfall model, also known as the traditional software development life cycle (SDLC).

1. The spiral model
2. Incremental process model
3. Agile development model

## Other models include:

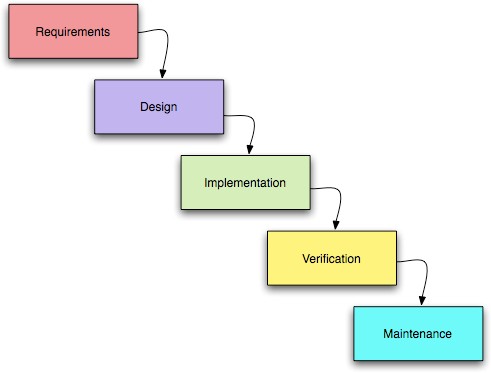
1. Build and fix model
2. The linear sequential model
3. The prototyping model (Rapid prototyping model)
4. Fountain model
5. Transformation model
6. Rapid Application Development (RAD) model
7. Evolutionary process model
8. Clean room software engineering model
9. Aspect-Oriented Software Development (AOSD) model

The software engineer chooses a specific software engineering model based on the nature of the project and application he/she is planning to develop. The methods and tools to be used, and the controls and deliverables are affected and influenced by the chosen model.

In our project we have use basic development model i.e Water Fall Model (SDLC)

This is also called Classic Life Cycle Model or Linear Sequential Model or Software Development Life Cycle Model (SDLC). This model has the following activities:

* 1. System Information Engineering and Modeling
  2. Software Requirement Analysis
  3. System Analysis and Design
  4. Code Generation
  5. Testing
  6. Maintenance



## System/Information Engineering and Modelling

As software development is large process so work begins by establishing requirements for all system elements and then allocating some subset of these requirements to software. The view of this system is necessary when software must interface with other elements such as hardware, people and other resources. System is the very essential requirement for the existence of software in any entity. In some cases for maximum output, the system should be re-engineered and spruced up. Once the ideal system is designed according to requirement, the development team studies the software requirement for the system.

## Software Requirement Analysis

Software Requirement Analysis is also known as feasibility study. In this requirement analysis phase, the development team visits the customer and studies their system requirement. They examine the need for possible software automation in the given software system. After feasibility study, the development team provides a document that holds the different specific recommendations for the candidate system. It also consists of personnel assignments, costs of the system, project schedule and target dates.

## System Analysis and Design

In this phase, the whole software development process, the overall software structure and its outlay are defined. In case of the clients/server processing technology, the number of tiers required for the package architecture, the database design, the data structure design etc are all defined. After designing part a software development model is created. Analysis and Design are very important in the whole development cycle process. Any fault in the design phase could be very expensive to solve in the software development process. In this phase, the logical system of the system product is developed.

## Code Generation

In Code Generation phase, the design must be decoded into a machine-readable form. If the design of software product is done in a detailed manner, code generation can be achieved without much complication.

## Testing

After code generation phase the software program testing begins. Different testing methods are available to detect the bugs that were committed during the previous phases. A number of testing tools and methods are already available for testing purpose.

## Maintenance

Software will definitely go through change once when it is delivered to the customer. There are large numbers of reasons for the change. Change could happen due to some unpredicted input values into the system. In addition to this the changes in the system directly have an effect on the software operations. The software should be implemented to accommodate changes that could be happen during the post development period.

## Modularization details

**MODULES:**

This project is aimed at developing an online Inventory Management System for a departmental

store.

The Project has the following modules

## Admin

**Visitor**

* A **Visitors** should be able to A visitor can visit website.

Visitor can check all categories with its sub category and its packages. Visitors can send suitable inquiry when they want to visit the place.

* The **administrator** should be able to

Login to the system and change his password after logging in.

Add, update, and delete users to the system.

Add, update, and delete categories.

Add, update, and delete subcategory.

Add, update, and delete packages.

Inquiry Confirmation of visitors.

Generates reports.

Checking billing activity.

## Process Flow Diagram

* + - * 1. **Analysis Model**

Analysis model is one of the most important parts of SRS. All the models give us clear view of the developing system. In this section, here is the list of all analysis models used in developing specific requirements of the current developing syst

## Scenario based model:

* Use Case Diagram.
* Activity Diagram.
* Swim lane Diagram.

## Flow Model

o Data Flow Diagram.

## Class Model:

* Class Diagram.
* CRC Diagram.

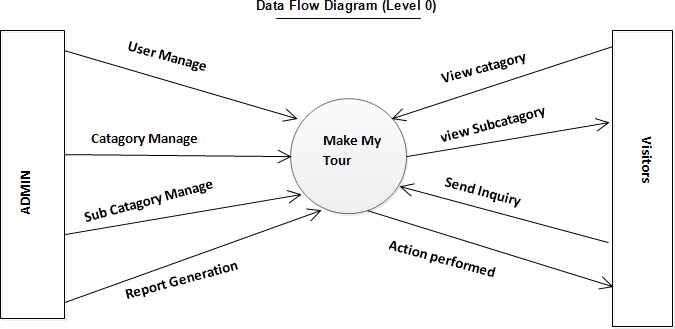
## Behavioral model:

* State transition Diagram.
* Sequence Diagram.
* Collaboration Diagram.

## Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing.

A DFD shows what kinds of data will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

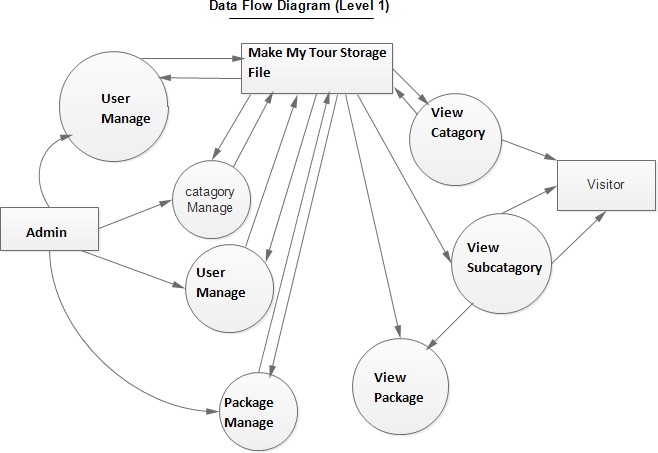


**Figure :** Data Flow Diagram (Level 0)

## Data flow diagram (level 0):

DFD level 0 , is the representation of the system which can only shows the inputs and outputs of the system without dealing with any function and file or database issue. In the given figure

this is the data flow diagram (Level 0) for our proposed Make My Tour. Control panel and the planning is two input entity which can give data command to the system, all the inputs will process in the system and the outputs will view in the display panel or as notification. And the system also can produce check sheet according to given planning.



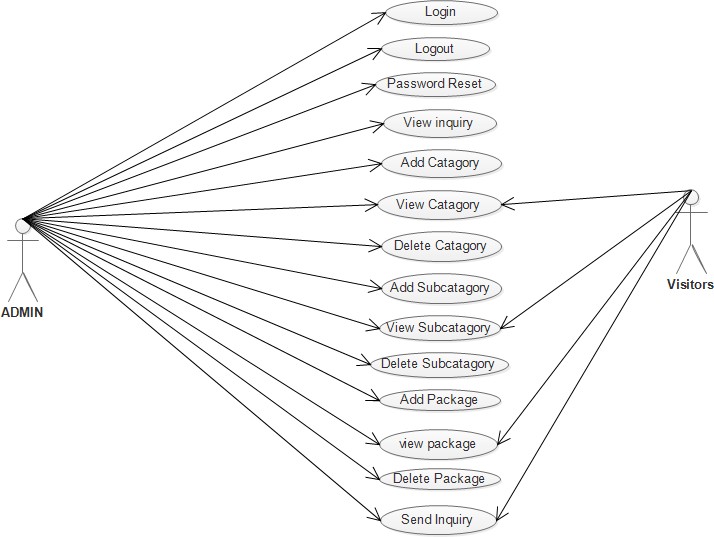
**Figure 6:** Data Flow Diagram (Level 1)

## Data flow diagram (Level 1):

DFD level 1, is the representation of the system which can visualize the relation among the functions and the file or database with the inputs and outputs. Control panel and planning entity can give input command to the system, which can process by some functions in this system like interact with user, configure, and update input, display status. There is a database related with configure function which one can modify production information. And display status, notification, check sheet can produce output depend on all the functions which configure inputs.

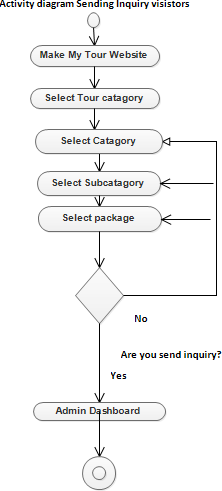
## Use Case diagram

In software and systems engineering, a use case is a list of steps, typically defining interactions between a role or actor and a system, to achieve a goal. The actor can be a human or an external system. Initially after analysing the requirements of our client we notify three potential actors in our proposed system. But this can change any time when their requirements are modified.

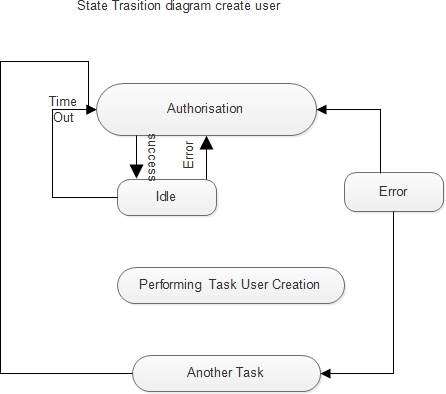


## Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

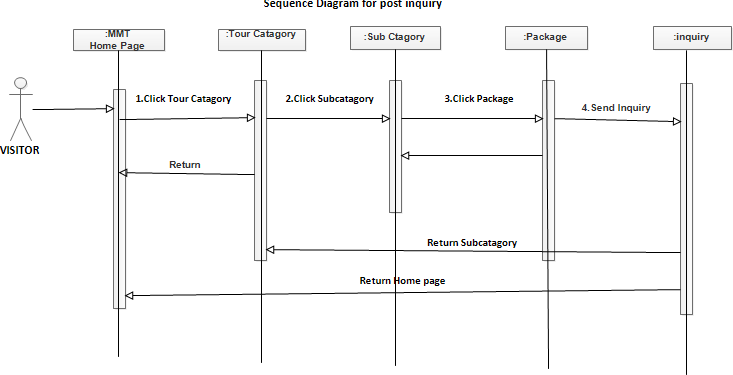


## State Transition Diagram

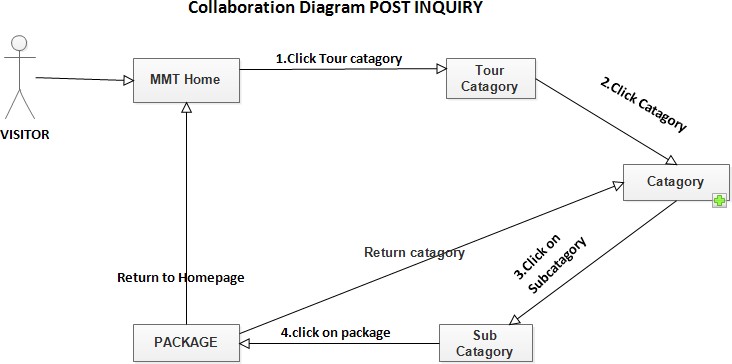
In the given figure, this is the state representation diagram for our proposed system. In the given figure there are four states, first one “reading state” just read the instructions. Second one “comparing state” compares the given command. When the state compare password, if the password is invalid, total system will be locked for a while. Or if the password is valid it will move a step ahead to select operation what the system allow an user.

## Sequence Diagram

A sequence diagram in a Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of our proposed system’s Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of Command exchanged between the objects needed to carry out the functionality of the scenario.



## Collaboration Diagram

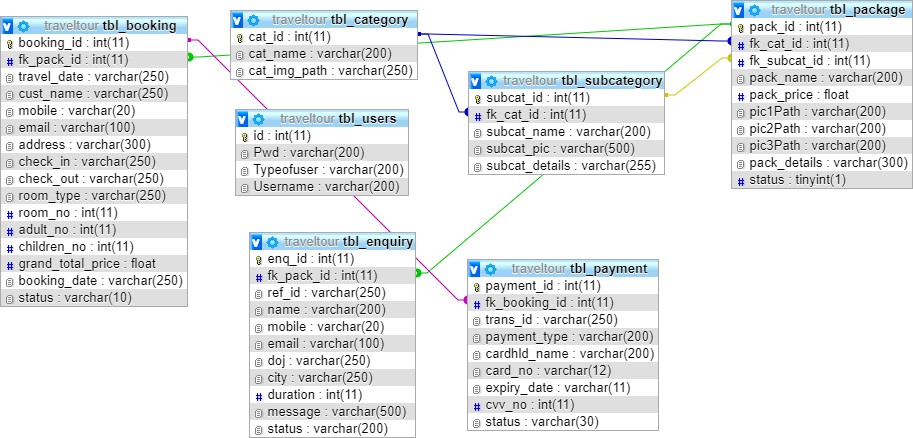


Collaboration diagrams belong to a group of UML diagrams called Interaction Diagrams. Collaboration diagrams, like Sequence Diagrams, show how objects interact over the course of time. However, instead of showing the sequence of events by the layout on the diagram, collaboration diagrams show the sequence by numbering the Commands on the diagram. This makes it easier to show how the objects are linked together, but harder to see the sequence at a glance**.**

## Database Design

The purpose of the IMS database system is to maintain the data that is used and generated from the warehouse staff and sales staff. Then the data stored will be used to facilitate smooth running of sales operation and stock management. The sole aim and objective of this thesis was to create a logical data model independent of Relational Database Management System for Online Inventory Management System. All the data generated during the business process was studied to design a very well-functioning database that will comfort the growth of company. These needs and requirements were given high importance throughout the design process. A system independent logical databasedesign was produced fromthis project work. The data dictionary, ER diagrams, conceptual data model, high level data transaction details, data flow diagrams and logical data model are the outcome of this project work.**.**

## ER Diagram



It is an abstract and conceptual representation of data. Entity-relationship modeling is a database modelling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database.

## Data Table

**2. 1.TBL\_CATEGORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **TYPE** | **KEY** | **EXTRA** |
| Cat\_id | Int(11) | Pri | Auto Incriment |
| Cat\_name | Varchar(200) |  |  |

## TBL\_SUBCATEGORY

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **TYPE** | **KEY** | **EXTRA** |
| Subcat\_id | Int(11) | Pri | Auto incriment |
| Fk\_cat\_id | Int(11) | Mul |  |
| Subcat\_name | Varchar(200) |  |  |
| Subcat\_pic | Varchar(500) |  |  |
| Subcat\_details | Varchar(255) |  |  |

1. **TBL\_PACKAGE**

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **TYPE** | **KEY** | **EXTRA** |
| Pack\_id | Int(11) | Pri | Auto incriment |
| Fk\_cat\_id | Int(11) | Mul |  |
| Fk\_subcat\_id | Int(11) | Mul |  |
| Pack\_name | Varchar(200) |  |  |
| Pack\_price | float |  |  |
| Pic1path | Varchar(200) |  |  |
| Pic2path | Varchar(200) |  |  |
| Pic3path | Varchar(200) |  |  |
| Pic\_details | Varchar(300) |  |  |
| Status | Tinyint(1) |  |  |

## TBL\_ENQUIRY

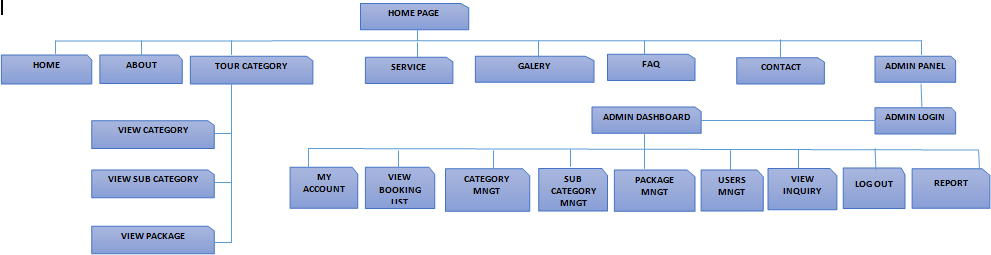
|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **TYPE** | **KEY** | **EXTRA** |
| Enq\_id | Int(11) | Pri | Auto incriment |

|  |  |  |  |
| --- | --- | --- | --- |
| Fk\_pack\_id | Int(11) | Mul |  |
| Name | Varchar(200) |  |  |
| Gender | Varchar(20) |  |  |
| Mobile | Varchar(20) |  |  |
| Email | Varchar(100) |  |  |
| Days | Int(20) |  |  |
| Child | Int(20) |  |  |
| Adult | Int(20) |  |  |
| Message | Varchar(500) |  |  |
| Details | Varchar(200) |  |  |

1. **TBL\_USERS**

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **TYPE** | **KEY** | **EXTRA** |
| Id | Int(11) | Pri | Auto incriment |
| Pwd | Varchar(200) |  |  |
| Typesofuser | Varchar(200) |  |  |
| Username | Varchar(200) |  |  |

## SITEMAP



**CHAPTER – 3**

## TOOLS AND TECHNOLOGY USED

* 1. **Development Tools:**

## Eclipse IDE:

Eclipse is an open-source software framework written primarily in Java. In its default form it is an Integrated Development Environment (IDE) for Java developers, consisting of the Java Development Tools (JDT) and the Eclipse Compiler for Java (ECJ). Users can extend its capabilities by installing plug-ins written for the Eclipse software framework, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules. Language packs are available for over a dozen languages.

## Architecture:

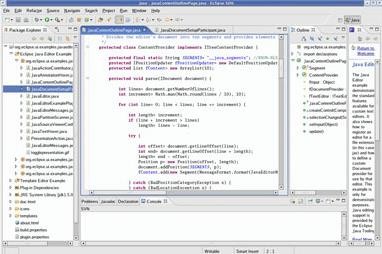
The basis for Eclipse is the Rich Client Platform (RCP). The following components constitute the rich client platform:

* + - * OSGi - a standard bundling framework
      * Core platform - boot Eclipse, run plug-ins
      * the Standard Widget Toolkit (SWT) - a portable widget toolkit
      * JFace - viewer classes to bring model view controller programming to SWT, file buffers, text handling, text editors
      * the Eclipse Workbench - views, editors, perspectives, wizards

Eclipse's widgets are implemented by a widget toolkit for Java called SWT, unlike most Java applications, which use the Java standard Abstract Window Toolkit (AWT) or Swing. Eclipse's user interface also leverages an intermediate GUI layer called JFace, which simplifies the construction of applications based on SWT.

Eclipse employs plug-ins in order to provide all of its functionality on top of (and including) the rich client platform, in contrast to some other applications where functionality is typically hard coded. This plug-in mechanism is a lightweight software componentry framework. In addition to allowing Eclipse to be extended using other programming languages such as C and Python, the plug-in framework allows Eclipse to work with typesetting languages like LaTeX, networking applications such as telnet, and database management systems. The plug-in architecture supports writing any desired extension to the environment, such as for configuration management. Java and CVS support is provided in the Eclipse SDK.

The key to the seamless integration of tools with Eclipse is the plug-in. With the exception of a small run-time kernel, everything in Eclipse is a plug-in. This means that a plug-in you develop integrates with Eclipse in exactly the same way as other plug-ins; in this respect, all features are created equal. The Eclipse SDK includes the Eclipse Java Development Tools, offering an IDE with a built-in incremental Java compiler and a full model of the Java source files. This allows for advanced refactoring techniques and code analysis. The IDE also makes use of a workspace, in this case set of metadata over a flat file space allowing external file modifications as long as the corresponding workspace "resource" is refreshed afterwards. The Visual Editor project allows interfaces to be created interactively, hence allowing Eclipse to be used as a RAD tool.



**The following is a list of notable projects and plug-in for the Eclipse IDE.**

These projects are maintained by the Eclipse community and hosted by the Eclipse Foundation.

## Core projects:

Rich Client Platform (Platform) is the core framework that all other Eclipse projects are built on.

Java Development Tools (JDT) provides support for core Java SE. This includes a standalone fast incremental compiler.

## Tools projects:

C/C++ Development Tools (CDT) adds support for C/C++ syntax highlighting, code formatting, and debugger integration and project structures. Unlike the JDT project, the CDT project does not add a compiler and relies on an external tool chain.

Graphical Editing Framework (GEF) allows developers to build standalone graphical tools. Example use includes circuit diagram design tools, activity diagram editors and WYSIWYG document editors.

## Web projects:

J2EE Standard Tools (JST) extends the core JDT to include support for Java EE projects. This includes EJBs, JSPs and Servlets.

PHP Development Tools (PDT)

Web Standard Tools (WST) adds standards compliant web development tools. These tools include editors for XML, HTML and CSS.

## Modeling projects:

Eclipse Modeling Framework (EMF) a modeling framework and code generation facility for building tools and other applications based on a structured data model, from a model specification described in XMI.

Graphical Modeling Framework (GMF) is a generative component and runtime infrastructure for developing graphical editors based on EMF and GEF.

## Other projects:

Test and Performance Tools Platform (TPTP) which provides a platform that allows software developers to build test and performance tools, such as debuggers, profilers and benchmarking applications.

Business Intelligence and Reporting Tools Project (BIRT), an Eclipse-based open source reporting system for web applications, especially those based on Java EE.

## Web Server(Apache Tomcat):

An Web server is a software engine that delivers applications to client computers or devices, typically through the Internet and using the Hypertext Transfer Protocol. Apache tomcat is an open source web container. It is not an application server like JBoss, Glassfish, etc. which are basically required for enterprise web applications. Typically consisting of EJBs and other heavy J2EE components.

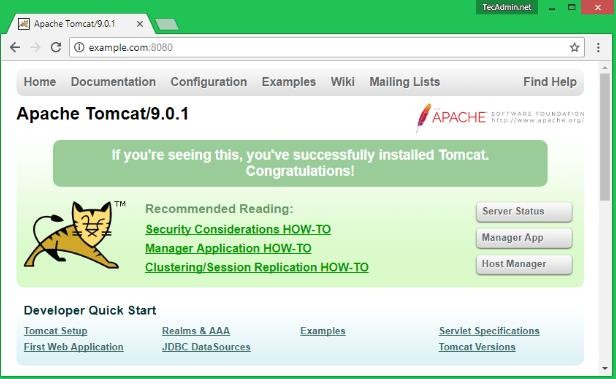
Tomcat is one of the most widely used web servers by java developers. Its current version is 8.x. It is used to run the web applications. It is used to running the following applications:

* 1. JSP/Servlet applications.
  2. Spring Framework based applications.
  3. Struts 2 applications.

## Common Features:

1. Use of a nonce to prevent cross-site request forgery (CSRF) attacks
2. Changing the jsessionid on authentication to prevent session fixation attacks altogether
3. Memory leak detection and prevention
4. Use of aliases to store static content outside the war file
5. Servlet 3.0, JSP 2.2 and JSP-EL 2.2 support
6. Easier to embed Tomcat in your applications, e.g. in JBoss
7. Asynchronous logging

According to Mark Thomas, Release Manager and Committer for Tomcat 9, the three most compelling features of Tomcat 9 are Servlet 3.0, memory leak prevention and detection, and improved security.



## Technology Used :

* + 1. **Server Side Technology**

## Java

Initially the language was called as “oak” but it was renamed as “Java” in 1995. The primary motivation of this language was the need for a platform-independent (i.e., architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

* Java is a programmer’s language.
* Java is cohesive and consistent.
* Except for those constraints imposed by the Internet environment, Java gives the programmer, full control.
* Finally, Java is to Internet programming where C was to system programming.

Java has had a profound effect on the Internet. This is because; Java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the Server and the Personal computer. They are: Passive information and Dynamic active programs. The Dynamic, Self-executing programs cause serious problems in the areas of Security and probability.

## Feature of Java Platform Independent:

The concept of Write-once-run-anywhere (known as the Platform independent) is one of the important key feature of java language that makes java as the most powerful language. Not even a single language is idle to this feature but java is closer to this feature. The programs written on one platform can run on any platform provided the platform must have the JVM.

**Simple:**

There are various features that make the java as a simple language. Programs are easy to write and debug because java does not use the pointers explicitly. It is much harder to write the java programs that can crash the system but we can not say about the other programming languages. Java provides the bug free system due to the strong memory management. It also has the automatic memory allocation and de-allocation system.

**Object Oriented:**

To be an Object Oriented language, any language must follow at least the four characteristics.

* + Inheritance : It is the process of creating the new classes and using the behavior of the existing classes by extending them just to reuse the existing code and adding the additional features as needed.
  + Encapsulation: It is the mechanism of combining the information and providing the abstraction.
  + Polymorphism: As the name suggest one name multiple form, Polymorphism is the way of providing the different functionality by the functions having the same name based on the signatures of the methods.
  + Dynamic binding: Sometimes we don't have the knowledge of objects about their specific types while writing our code. It is the way of providing the maximum functionality to a program about the specific type at runtime.

As the languages like Objective C, C++ fulfills the above four characteristics yet they are not fully object oriented languages because they are structured as well as object oriented languages. But in case of java, it is a fully Object Oriented language because object is at the outer most level of data structure in java. No stand alone methods, constants, and variables are there in java. Everything in java is object even the primitive data types can also be converted into object by using the wrapper class.

**Robust:**

Java has the strong memory allocation and automatic garbage collection mechanism. It provides the powerful exception handling and type checking mechanism as compare to other programming languages. Compiler checks the program whether there any error and interpreter checks any run time error and makes the system secure from crash. All of the above features makes the java language robust.

**Distributed:**

The widely used protocols like HTTP and FTP are developed in java. Internet programmers can call functions on these protocols and can get access the files from any remote machine on the internet rather than writing codes on their local system.

**Portable:**

The feature Write-once-run-anywhere makes the java language portable provided that the system must have interpreter for the JVM. Java also have the standard data size irrespective of operating system or the processor. These features make the java as a portable language.

**Dynamic:**

While executing the java program the user can get the required files dynamically from a local drive or from a computer thousands of miles away from the user just by connecting with the Internet.

**Secure:**

Java does not use memory pointers explicitly. All the programs in java are run under an area known as the sand box. Security manager determines the accessibility options of a class like reading and writing a file to the local disk. Java uses the public key encryption system to allow the java applications to transmit over the internet in the secure encrypted form. The byte code Verifier checks the classes after loading.

**Performance:**

Java uses native code usage, and lightweight process called threads. In the beginning interpretation of byte code resulted the performance slow but the advance version of JVM uses the adaptive and just in time compilation technique that improves the performance.

**Multithreaded:**

Java is also a multithreaded programming language. Multithreading means a single program having different threads executing independently at the same time. Multiple threads execute instructions according to the program code in a process or a program. Multithreading works the similar way as multiple processes run on one computer. Multithreading programming is a very interesting concept in Java. In multithreaded programs not even a single thread disturbs the execution of other thread. Threads are obtained from the pool of available ready to run threads and they run on the system CPUs. This is how Multithreading works in Java which you will soon come to know in details in later chapters.

**Interpreted:**

we all know that Java is an interpreted language as well. With an interpreted language such as Java, programs run directly from the source code. The interpreter program reads the source code and translates it on the fly into computations. Thus, Java as an interpreted language depends on an interpreter program. The versatility of being **platform independent** makes Java to outshine from other languages. The source code to be written and distributed is platform independent. Another advantage of Java as an interpreted language is its error debugging quality. Due to this any error occurring in the program gets traced. This is how it is different to work with Java.

**Architecture Neutral:**

The term architectural neutral seems to be weird, but yes Java is an architectural neutral language as well. The growing popularity of networks makes developers think distributed. In the world of network it is essential that the applications must be able to migrate easily to different computer systems. Not only to computer systems but to a wide variety of hardware

architecture and operating system architectures as well. The Java compiler does this by generating byte code instructions, to be easily interpreted on any machine and to be easily translated into native

machine code on the fly.The compiler generates an architecture-neutral object file format to enable a Java application to execute anywhere on the network and then the compiled code is executed on many processors, given the presence of the Java runtime system.Hence Java was designed to support applications on network. This feature of Java has thrived the programming language.

**ABOUT : JDK:**

The **Java Development Kit** (**JDK**) is a [Sun Microsystems](http://en.wikipedia.org/wiki/Sun_Microsystems) product aimed at [Java](http://en.wikipedia.org/wiki/Java_(programming_language)) developers. Since the introduction of Java, it has been by far the most widely used Java [SDK](http://en.wikipedia.org/wiki/Software_development_kit). On [17 November](http://en.wikipedia.org/wiki/November_17)[2006](http://en.wikipedia.org/wiki/2006), Sun announced that it would be released under the [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License) (GPL), thus making it [free software.](http://en.wikipedia.org/wiki/Free_software) This happened in large part on [8 May](http://en.wikipedia.org/wiki/May_8)[2007](http://en.wikipedia.org/wiki/2007)[[1]](http://en.wikipedia.org/wiki/Java_Development_Kit#cite_note-0%23cite_note-0) and the source code was contributed to the [OpenJDK](http://en.wikipedia.org/wiki/OpenJDK).

The primary components of the JDK are a selection of programming tools, including:

* + - * 1. java – The [loader](http://en.wikipedia.org/wiki/Loader) for Java applications. This tool is an interpreter and can interpret the class files generated by the [javac](http://en.wikipedia.org/wiki/Javac) compiler. Now a single launcher is used for both development and deployment. The old deployment launcher, jre, is no longer provided with Sun JDK.
        2. [javac](http://en.wikipedia.org/wiki/Javac) – The [compiler,](http://en.wikipedia.org/wiki/Compiler) which converts source code into [Java bytecode](http://en.wikipedia.org/wiki/Java_bytecode)
        3. jar – The archiver, which packages related class [libraries](http://en.wikipedia.org/wiki/Library_(computer_science)) into a single [JAR file](http://en.wikipedia.org/wiki/Jar_(file_format)). This tool also helps manage JAR files.
        4. [javadoc](http://en.wikipedia.org/wiki/Javadoc) – The documentation generator, which automatically generates documentation from [source code](http://en.wikipedia.org/wiki/Source_code) comments
        5. jdb – The [debugger](http://en.wikipedia.org/wiki/Debugger)
        6. javap – The class file disassembler
        7. appletviewer – This tool can be used to run and debug Java applets without a web browser.
        8. javah – The C header and stub generator, used to write native methods
        9. extcheck – This utility can detect JAR-file conflicts.
        10. apt – The annotation processing tool
        11. jhat – (Experimental) Java heap analysis tool
        12. jstack – (Experimental) This utility prints Java stack traces of Java threads.
        13. jstat – (Experimental) [Java Virtual Machine](http://en.wikipedia.org/wiki/Java_Virtual_Machine) statistics monitoring tool
        14. jstatd – (Experimental) jstat daemon
        15. jinfo – (Experimental) This utility gets configuration information from a running Java process or crash dump.
        16. jmap – (Experimental) This utility outputs the memory map for Java and can print shared object memory maps or heap memory details of a given process or core dump.
        17. idlj – The IDL-to-Java compiler. This utility generates Java bindings from a given IDL file.
        18. policy tool – The policy creation and management tool, which can determine policy for a Java runtime, specifying which permissions are available for code from various sources
        19. [VisualVM](http://en.wikipedia.org/wiki/VisualVM) – visual tool integrating several command line JDK tools and lightweight performance and memory profiling capabilities

The JDK also comes with a complete [Java Runtime Environment](http://en.wikipedia.org/wiki/Java_Runtime_Environment), usually called a *private* runtime. It consists of a [Java Virtual Machine](http://en.wikipedia.org/wiki/Java_Virtual_Machine) and all of the class libraries that will be present in

## Java Server Page(JSP)

Java server Pages is a simple, yet powerful technology for creating and maintaining dynamic- content web pages. Based on the Java programming language, Java Server Pages offers proven portability, open standards, and mature re-usable component model .The Java Server Pages architecture enables the separation of content generation from content presentation. This separation not eases maintenance headaches; it also allows web team members to focus on their areas of expertise. Now, web page designer can concentrate on layout, and web application designers on programming, with minimal concern about impacting each other’s work.

## Features of JSP Portability:

Java Server Pages files can be run on any web server or web-enabled application server that provides support for them. Dubbed the JSP engine, this support involves recognition, translation, and management of the Java Server Page lifecycle and its interaction components.

## Components

It was mentioned earlier that the Java Server Pages architecture can include reusable Java components. The architecture also allows for the embedding of a scripting language directly into the Java Server Pages file. The components current supported include Java Beans, and Servlets.

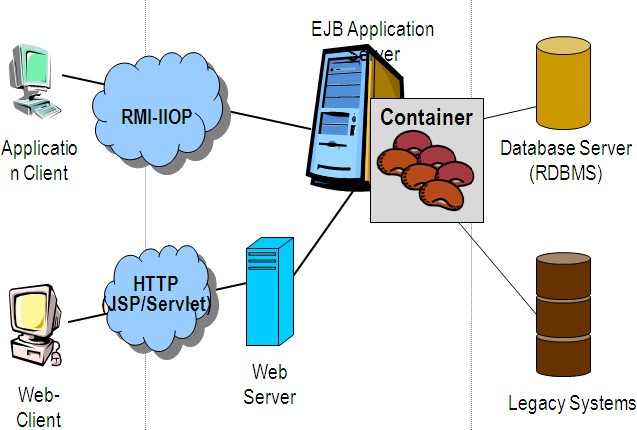
## Processing

A Java Server Pages file is essentially an HTML document with JSP scripting or tags. The Java Server Pages file has a JSP extension to the server as a Java Server Pages file. Before the page is served, the Java Server Pages syntax is parsed and processed into a Servlet on the server side. The Servlet that is generated outputs real content in straight HTML for responding to the client.

## Access Models:

A Java Server Pages file may be accessed in at least two different ways. A client’s request comes directly into a Java Server Page. In this scenario, suppose the page accesses reusable Java Bean components that perform particular well-defined computations like accessing a database. The result of the Beans computations, called result sets is stored within the Bean as properties. The page uses such Beans to generate dynamic content and present it back to the client.

In both of the above cases, the page could also contain any valid Java code. Java Server Pages architecture encourages separation of content from presentation.



## Steps in the execution of a JSP Application:

* The client sends a request to the web server for a JSP file by giving the name of the JSP file within the form tag of a HTML page.
* This request is transferred to the JavaWebServer. At the server side JavaWebServer receives the request and if it is a request for a jsp file server gives this request to the JSP engine.
* JSP engine is program which can under stands the tags of the jsp and then it converts those tags into a Servlet program and it is stored at the server side. This Servlet is loaded in the memory and then it is executed and the result is given back to the JavaWebServer and then it is transferred back to the result is given back to the JavaWebServer and then it is transferred back to the client.

## JDBC

JDBC is a Java API for executing SQL statements. (As a point of interest, JDBC is a trademarked name and is not an acronym; nevertheless, JDBC is often thought of as standing for Java Database Connectivity. It consists of a set of classes and interfaces written in the Java programming language. JDBC provides a standard API for tool/database developers and makes it possible to write database applications using a pure Java API.

Using JDBC, it is easy to send SQL statements to virtually any relational database. One can write a single program using the JDBC API, and the program will be able to send SQL statements to the appropriate database. The combinations of Java and JDBC lets a programmer write it once and run it anywhere.

What Does JDBC Do?

Simply put, JDBC makes it possible to do three things:

* Establish a connection with a database
* Send SQL statements
* Process the results. JDBC versus ODBC and other APIs

At this point, Microsoft's ODBC (Open Database Connectivity) API is that probably the most widely used programming interface for accessing relational databases. It offers the ability to connect to almost all databases on almost all platforms.

So why not just use ODBC from Java? The answer is that you can use ODBC from Java, but this is best done with the help of JDBC in the form of the JDBC-ODBC Bridge, which we will cover shortly. The question now becomes "Why do you need JDBC?" There are several answers to this question:

* ODBC is not appropriate for direct use from Java because it uses a C interface. Calls from Java to native C code have a number of drawbacks in the security, implementation, robustness, and automatic portability of applications.
* A literal translation of the ODBC C API into a Java API would not be desirable. For example, Java has no pointers, and ODBC makes copious use of them, including the notoriously error-prone generic pointer "void \*". You can think of JDBC as ODBC translated into an object-oriented interface that is natural for Java programmers.
* ODBC is hard to learn. It mixes simple and advanced features together, and it has complex options even for simple queries. JDBC, on the other hand, was designed to keep simple things simple while allowing more advanced capabilities where required.
* A Java API like JDBC is needed in order to enable a "pure Java" solution. When ODBC is used, the ODBC driver manager and drivers must be manually installed on every client machine. When the JDBC driver is written completely in Java, however, JDBC code is automatically installable, portable, and secure on all Java platforms from network computers to mainframes.

Two-tier and Three-tier Models

The JDBC API supports both two-tier and three-tier models for database access.

In the two-tier model, a Java applet or application talks directly to the database. This requires a JDBC driver that can communicate with the particular database management system being accessed. A user's SQL statements are delivered to the database, and the results of those statements are sent back to the user. The database may be located on another machine to which the user is connected via a network. Thisis referred to as a client/server configuration, with the user's machine as the client, and the machine housing the database as the server. The network can be an Intranet, which, for example, connects employees within a corporation, or it can be the Internet.

**Client HTTP, RMI, or**

**Applicati on Server (Java)**

**Java applet or Html**

**Server machine DBMS-**

**Database**

35

**Client**

JDBC

**JAVA**

**Applicat**

**DBMS-**

**proprietary**

**Databas**

DBMS

In the three-tier model, commands are sent to a "middle tier" of services, which then send SQL statements to the database. The database processes the SQL statements and sends the results back to the middle tier, which then sends them to the user. MIS directors find the three-tier model very attractive because the middle tier makes it possible to maintain control over access and the kinds of updates that can be made to corporate data. Another advantage is that when there is a middle tier, the user can employ an easy-to-use higher-level API which is translated by the middle tier into the appropriate low-level calls. Finally, in many cases the three-tier architecture can provide performance advantages.

Until now the middle tier has typically been written in languages such as C or C++, which offer fast performance. However, with the introduction of optimizing compilers that translate Java byte code into efficient machine-specific code, it is becoming practical to implement the middle tier in Java. This is a big plus, making it possible to take advantage of Java's robustness, multithreading, and security features. JDBC is important to allow database access from a Java middle tier.

## JDBC Driver Types

The JDBC drivers that we are aware of at this time fit into one of four categories:

* + - * 1. JDBC-ODBC bridge plus ODBC driver
        2. Native-API partly-Java driver
        3. JDBC-Net pure Java driver
        4. Native-protocol pure Java driver

## Client side Technology

* + - 1. **HTML**

HTML, an initialism of Hypertext Markup Language, is the predominant markup language for web pages. It provides a means to describe the structure of text-based information in a document — by denoting certain text as headings, paragraphs, lists, and so on — and to supplement that text with interactive forms, embedded images, and other objects. HTML is written in the form of labels (known as tags), surrounded byangle brackets. HTML can also describe, to some degree, the appearance and semantics of a document, and can include embedded scripting language code which can affect the behavior of web browsers and other HTML processors.

HTML is also often used to refer to content of the MIME type text/html or even more broadly as a generic term for HTML whether in its XML-descended form (such as XHTML 1.0 and later) or its form descended directly from SGML

Hyper Text Markup Language

Hypertext Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produces Web pages that include text, graphics and pointer to other Web pages (Hyperlinks).

HTML is not a programming language but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language), but specialized to hypertext and adapted to the Web. The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point. We can navigate through the information based on our interest and preference. A markup language is simply a series of elements, each delimited with special characters that define how text or other items enclosed within the elements should be displayed. Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document.

HTML can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop.

HTML provides tags (special codes) to make the document look attractive. HTML tags are not case- sensitive. Using graphics, fonts, different sizes, color, etc., can enhance the presentation of the document. Anything that is not a tag is part of the document itself.

Basic HTML Tags:

<! -- --> specifies comments

<A>……….</A> Creates hypertext links

<B>……….</B> Formats text as bold

<BIG>……….</BIG> Formats text in large font.

<BODY>…</BODY> Contains all tags and text in the HTML document

<CENTER>...</CENTER> Creates text

<DD>…</DD> Definition of a term

<DL>...</DL> Creates definition list

<FONT>…</FONT> Formats text with a particular font

<FORM>...</FORM>Encloses a fill-out form

<FRAME>...</FRAME> Defines a particular frame in a set of frames

<H#>…</H#>Creates headings of different levels( 1 – 6 )

<HEAD>...</HEAD> Contains tags that specify information about a document

<HR>...</HR> Creates a horizontal rule

<HTML>…</HTML> Contains all other HTML tags

<META>...</META>Provides meta-information about a document

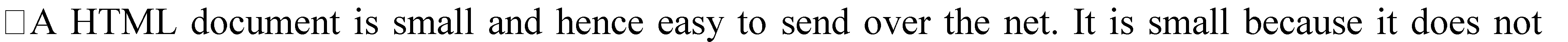
<SCRIPT>…</SCRIPT> Contains client-side or server-side script

<TABLE>…</TABLE> Creates a table

<TD>…</TD> Indicates table data in a table

<TR>…</TR> Designates a table row

<TH>…</TH> Creates a heading in a table Advantages



include formatted information.

HTML is platform independent. HTML tags are not case-sensitive.



## JavaScript

JavaScript is a script-based programming language that was developed by Netscape Communication Corporation. JavaScript was originally called Live Script and renamed as JavaScript to indicate its relationship with Java. JavaScript supports the development of both client and server components of

Web-based applications. On the client side, it can be used to write programs that are executed by a Web browser within the context of a Web page. On the server side, it can be used to write Web server programs that can process information submitted by a Web browser and then update the browser’s display accordingly.

Even though JavaScript supports both client and server Web programming, we prefer JavaScript at Client side programming since most of the browsers supports it. JavaScript is almost as easy to learn as HTML, and JavaScript statements can be included in HTML documents by enclosing the statements between a pair of scripting tags<SCRIPTS>..</SCRIPT>.

<SCRIPT LANGUAGE = “JavaScript”>

JavaScript statements

</SCRIPT>

Here are a few things we can do with JavaScript:

Validate the contents of a form and make calculations.



Add scrolling or changing messages to the Browser’s status line.

Animate images or rotate images that change when we move the mouse over them. Detect the browser in use and display different content for different browsers.

Detect installed plug-ins and notify the user if a plug-in is required. We can do much more with JavaScript, including creating entire application. JavaScript Vs Java

JavaScript and Java are entirely different languages. A few of the most glaring differences are:

* Java applets are generally displayed in a box within the web document; JavaScript can affect any part of the Web document itself.
* While JavaScript is best suited to simple applications and adding interactive features to Web pages; Java can be used for incredibly complex applications.

## Advantages

JavaScript can be used for Sever-side and Client-side scripting. It is more flexible than VBScript.



JavaScript is the default scripting languages at Client-side since all the browsers supports it.

## CSS

**C**ascading **S**tyle **S**heets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs,variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

**Advantages of CSS**

* **CSS saves time** − You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
* **Pages load faster** − If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
* **Easy maintenance** − To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
* **Superior styles to HTML** − CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
* **Multiple Device Compatibility** − Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
* **Global web standards** − Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers

## Database Server

MySQL is a fast, easy to use relational database. It is currently the most popular open-source database. It is very commonly used in conjunction with PHP scripts to create powerful and dynamic server-side applications.

MySQL is used for many small and big businesses. It is developed, marketed and supported by MySQL AB, a Swedish company. It is written in C and C++.

MySQL is becoming so popular because of these following reasons:

* + - * MySQL is an open-source database so you don't have to pay a single penny to use it.
      * MySQL is a very powerful program so it can handle a large set of functionality of the most expensive and powerful database packages.
      * MySQL is customizable because it is an open source database and the open-source GPL license facilitates programmers to modify the SQL software according to their own specific environment.
      * MySQL is quicker than other databases so it can work well even with the large data set.
      * MySQL supports many operating systems with many languages like PHP, PERL, C, C++, JAVA, etc.
      * MySQL uses a standard form of the well-known SQL data language.
      * MySQL is very friendly with PHP, the most popular language for web development.
      * MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).

## CHAPTER – 4

**CODDING**

## IMPLEMENTATION PHASE

The implementation phase is less creative than system design. It is primarily concerned with user training, site preparation, and file conversion. When the candidate system is linked to terminals or remote sites, telecommunication network and test of the network along with the system are also included under implementation.

During the final testing, user acceptance is tested, followed by user training. Depending on the nature of the system, extensive user training may be required. Conversion usually takes place about the same time the user is being trained or late.

In the extreme, the programmer is viewed as someone who ought to be isolated from other aspects of the system development. Programming is itself design work, however. The initial parameters of the candidate system should be modified as a result of programming efforts. Programming provides a ‘reality test’ for the assumptions made by the analyst. Therefore, it is a mistake to exclude programmers from the initial system design.

## CODE SNIPPETS: A few important highlights of the code are unveiled below.

|  |
| --- |
| **JAVA imports for establishing DBConnection.java :** |
| **package** travel;  **import** java.sql.\*;  **publicclass** DBConnection {  **public** Connection Connect()  {  Connection con=**null**;  **try**  {  Class.*forName*("com.mysql.jdbc.Driver"); con=DriverManager.*getConnection*("jdbc:mysql://localhost:3306/inventory","root","");  }  **catch**(Exception e)  {  e.printStackTrace();  }  **return** con;  }  } |

|  |
| --- |
| **Code for bean Class Category.java :** |
| package travel;  public class Category {  private int cat\_id; private String cat\_name; public int getCat\_id() {  return cat\_id;  }  public void setCat\_id(int cat\_id) { this.cat\_id = cat\_id;  }  public String getCat\_name() { return cat\_name;  }  public void setCat\_name(String cat\_name) { this.cat\_name = cat\_name;  }  } |

|  |
| --- |
| **Code for DAO Class CategoryDao.java :** |
| package travel;  import java.sql.Connection;  import java.sql.PreparedStatement; import java.sql.ResultSet;  import java.util.ArrayList; import java.util.List; import travel.Category;  import travel.DBConnection; public class CategoryDao {  public static int save(Category u){ int status=0;  try{  Connection con= new DBConnection().Connect(); PreparedStatement ps=con.prepareStatement(  "insert into tbl\_category(cat\_name) values(?)"); ps.setString(1,u.getCat\_name()); status=ps.executeUpdate();  }catch(Exception e){System.out.println(e);} return status;  }  public static int update(Category u){ int status=0;  try{  Connection con= new DBConnection().Connect(); PreparedStatement ps=con.prepareStatement(  "update tbl\_category set cat\_name=? where cat\_id=?"); System.out.println(u.getCat\_name());  ps.setString(1,u.getCat\_name()); ps.setInt(2,u.getCat\_id()); status=ps.executeUpdate();  }catch(Exception e){System.out.println(e);} return status;  }  public static int delete(Category u){ int status=0;  try{  Connection con= new DBConnection().Connect();  PreparedStatement ps=con.prepareStatement("delete from tbl\_category where id=?"); ps.setInt(1,u.getCat\_id());  status=ps.executeUpdate();  }catch(Exception e){System.out.println(e);} |

|  |
| --- |
| return status;  }  public static List<Category> getAllRecords(){ List<Category> list=new ArrayList<Category>();  try{  Connection con= new DBConnection().Connect();  PreparedStatement ps=con.prepareStatement("select \* from tbl\_category"); ResultSet rs=ps.executeQuery();  while(rs.next()){  Category u=new Category(); u.setCat\_id(rs.getInt("cat\_id")); u.setCat\_name(rs.getString("cat\_name")); list.add(u);  }  }catch(Exception e){System.out.println(e);} return list;  }  public static Category getRecordById(int cat\_id){ Category u=null;  try{  Connection con= new DBConnection().Connect();  PreparedStatement ps=con.prepareStatement("select \* from tbl\_category where  cat\_id=?");  ps.setInt(1,cat\_id);  ResultSet rs=ps.executeQuery(); while(rs.next()){  u=new Category(); u.setCat\_id(rs.getInt("cat\_id")); u.setCat\_name(rs.getString("cat\_name"));  }  }catch(Exception e){System.out.println(e);} return u;  }  } |
| **HTML Code For Admin Login Page** |
| **admin\_login.html**  <!DOCTYPE html>  <html lang=*"en"*>  <head>  <meta charset=*"utf-8"*>  <meta http-equiv=*"X-UA-Compatible"* content=*"IE=edge"*>  <meta name=*"viewport"* content=*"width=device-width, initial-scale=1, shrink-to-fit=no"*>  <meta name=*"description"* content=*""*>  <meta name=*"author"* content=*""*> |

|  |
| --- |
| <title>Makemy Travel</title>  <!-- Custom fonts for this template-->  <link href=*"vendor/fontawesome-free/css/all.min.css"* rel=*"stylesheet"* type=*"text/css"*>  <link href=*"https://fonts.googleapis.com/css?family=Nunito:200,200i,300,300i,400,400i,600,600i,700,7 00i,800,800i,900,900i"* rel=*"stylesheet"*>  <!-- Custom styles for this template-->  <link href=*"css/sb-admin-2.css"* rel=*"stylesheet"*>  </head>  <body class=*"bg-gradient-info"*>  <div class=*"container"*>  <!-- Outer Row -->  <div class=*"row justify-content-center"*>  <div class=*"col-xl-10 col-lg-12 col-md-9"*>  <div class=*"card o-hidden border-0 shadow-lg my-5"*>  <div class=*"card-body p-0"*>  <!-- Nested Row within Card Body -->  <div class=*"row"*>  <div class=*"col-lg-6 d-none d-lg-block bg-login-image"*></div>  <div class=*"col-lg-6"*>  <div class=*"p-5"*>  <div class=*"text-center"*>  <h1 class=*"h4 font-weight-bold mb-4 text-danger"*>Admin Login !</h1>  </div>  <form class=*"user"* action=*"loginprocess.jsp"* method=*"post"*>  <div class=*"form-group"*>  <input type=*"text"* class=*"form-control form-control-user"* id=*"exampleInputUser"* aria- describedby=*"userlHelp"* placeholder=*"Username"* name=*"user"*>  </div>  <div class=*"form-group"*>  <input type=*"password"* class=*"form-control form-control-user"* id=*"exampleInputPassword"* placeholder=*"Password"* name=*"password"*>  </div>  <div class=*"form-group"*>  <div class=*"custom-control custom-checkbox small"*>  <input type=*"checkbox"* class=*"custom-control-input"* id=*"customCheck"*>  <label class=*"custom-control-label"* for=*"customCheck"*>Remember Me</label>  </div>  </div>  <button class=*"btn btn-primary btn-user btn-block"*> Login  </button> |

<hr>

<a href=*"#"* class=*"btn btn-google btn-user btn-block"*>

<i class=*"fab fa-google fa-fw"*></i> Login with Google

</a>

<a href=*"#"* class=*"btn btn-facebook btn-user btn-block"*>

<i class=*"fab fa-facebook-f fa-fw"*></i> Login with Facebook

</a>

</form>

<hr>

<div class=*"text-center"*>

<a class=*"small"* href=*"forgot-password.html"*>Forgot Password?</a>

</div>

<div class=*"text-center"*>

<a class=*"small"* href=*"register.html"*>Create an Account!</a>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

<!-- Bootstrap core JavaScript-->

<script src=*"vendor/jquery/jquery.min.js"*></script>

<script src=*"vendor/bootstrap/js/bootstrap.bundle.min.js"*></script>

<!-- Core plugin JavaScript-->

<script src=*"vendor/jquery-easing/jquery.easing.min.js"*></script>

<!-- Custom scripts for all pages-->

<script src=*"js/sb-admin-2.min.js"*></script>

</body>

</html>

|  |
| --- |
| **Controller for Package** |
| **PackageProcess.java**  package travel; |

import java.io.File;

import java.io.IOException; import java.io.PrintWriter; import java.util.Date;

import javax.servlet.ServletException;

import javax.servlet.annotation.MultipartConfig; import javax.servlet.annotation.WebServlet; import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest; import javax.servlet.http.HttpServletResponse; import javax.servlet.http.Part;

/\*\*

\* Servlet implementation class PackageProcess

\*/ @WebServlet("/PackageProcess")

@MultipartConfig(maxFileSize = 1024 \* 1024 \* 2) public class PackageProcess extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public PackageProcess() { super();

// TODO Auto-generated constructor stub

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.setContentType("text/html"); PrintWriter out = response.getWriter();

try{

int

fk\_cat\_id=Integer.parseInt(request.getParameter("fk\_cat\_id"));

System.out.println(fk\_cat\_id); int

fk\_subcat\_id=Integer.parseInt(request.getParameter("fk\_subcat\_id"));

System.out.println(fk\_subcat\_id);

String pack\_name=request.getParameter("pack\_name"); float

pack\_price=Float.parseFloat(request.getParameter("pack\_price"));

Part p1=request.getPart("pic1Path"); Part p2=request.getPart("pic2Path"); Part p3=request.getPart("pic3Path");

String pack\_details=request.getParameter("pack\_details");

//String subcat\_details=request.getParameter("subcat\_details");

// Getting Application Path

//String appPath = request.getServletContext().getContextPath(); String appPath ="E:/Eclipse/Travel/WebContent/PackageImages";

// File path where all files will be stored String imagePath = appPath + "";

// Creates the file directory if it does not exists File fileDir = new File(imagePath);

if (!fileDir.exists()) { fileDir.mkdirs();

}

//Get Image Name

String imageName1 = p1.getSubmittedFileName(); System.out.println(imageName1);

String fileExt1 = imageName1.substring(imageName1.length()-3); String imgname1=new Date().getTime() +"1"+"."+fileExt1;

String imageName2 = p2.getSubmittedFileName();

String fileExt2 = imageName2.substring(imageName2.length()-3); String imgname2=new Date().getTime() +"2"+"."+fileExt2;

String imageName3 = p3.getSubmittedFileName();

String fileExt3 = imageName3.substring(imageName3.length()-3); String imgname3=new Date().getTime() +"3"+"."+fileExt3;

//System.out.println(imagePath);

//System.out.println(imageName);

//System.out.println(imagePath + "/"+ imgname); String finalImgPath1=imagePath + "/"+ imgname1; String finalImgPath2=imagePath + "/"+ imgname2; String finalImgPath3=imagePath + "/"+ imgname3;

if(validateImage1(imageName1)&& validateImage2(imageName2) && validateImage3(imageName3)){

try{

p1.write(imagePath + "/" + imgname1); p2.write(imagePath + "/" + imgname2); p3.write(imagePath + "/" + imgname3);

//out.print("<img src=\"images/"+imageName+"\" >");

}catch (Exception ex) {

//out.print("<h1>"+ex.getMessage()+"</h1>");

}

}else{

out.print("<script> alert('Invalid Image Format')</script>");

}

/\*

PreparedStatement ps=con.

* Connection con= new DBConnection().Connect();
* prepareStatement("insert into

tbl\_subcategory(fk\_cat\_id,subcat\_name,subcat\_pic,subcat\_details) values(?,?,?,?)"

* + ); ps.setInt(1, 1); ps.setString(2,subcat\_name);
  + ps.setString(3,finalImgPath);

ps.setString(4,subcat\_details);

* status=ps.executeUpdate();

\*/

Package u=new Package(); u.setFk\_cat\_id(fk\_cat\_id);

u.setFk\_subcat\_id(fk\_subcat\_id); u.setPack\_name(pack\_name); u.setPack\_price(pack\_price); u.setPic1Path(finalImgPath1); u.setPic2Path(finalImgPath2); u.setPic3Path(finalImgPath3); u.setPack\_details(pack\_details);

int i=PackageDao.save(u); response.sendRedirect("package\_view.jsp");

}

catch(Exception e){

e.printStackTrace();

}

}

private boolean validateImage1(String imageName1){

String fileExt1 = imageName1.substring(imageName1.length()-3); if("jpg".equals(fileExt1)||"png".equals(fileExt1) || "gif".equals(fileExt1))

return true;

return false;

}

private boolean validateImage2(String imageName2){

String fileExt2 = imageName2.substring(imageName2.length()-3); if("jpg".equals(fileExt2)||"png".equals(fileExt2) || "gif".equals(fileExt2))

return true;

return false;

}private boolean validateImage3(String imageName3){

String fileExt3 = imageName3.substring(imageName3.length()-3); if("jpg".equals(fileExt3)||"png".equals(fileExt3) || "gif".equals(fileExt3))

return true;

return false;

}

}

|  |
| --- |
| **Code for viewinquiry.jsp :** |
| <!-- Begin Page Content -->  <div class=*"container-fluid"*>  <form method=*"post"*>  <table border=*"0"*width=*"90%"*height=*"300px"*align=*"center"* class=*"tableshadow"*>  <tr><td class=*"toptd"*>View Enquiry</td></tr>  <tr><td align=*"center"* valign=*"top"* style="padding-top:*10px*;">  <table border=*"0"*align=*"center"*width=*"95%"*>  <tr><td style="font-size:*15px*; padding:*5px*; font-weight:*bold*;" >Package Name</td>  <td style="font-size:*15px*; padding:*5px*; font-weight:*bold*; ">Package Id</td>  <td style="font-size:*15px*; padding:*5px*; font-weight:*bold*;">Name</td>  <td style="font-size:*15px*; padding:*5px*; font-weight:*bold*;">Gender</td>  <td style="font-size:*15px*; padding:*5px*; font-weight:*bold*;">Mobile No.</td>  <td style="font-size:*15px*; padding:*5px*; font-weight:*bold*;">Email</td>  <td style="font-size:*15px*; padding:*5px*; font-weight:*bold*;">No. of Days</td>  <td style="font-size:*15px*; padding:*5px*; font-weight:*bold*;">No. of Children</td>  <td style="font-size:*15px*; padding:*5px*; font-weight:*bold*;">no. of Adults</td>  <td style="font-size:*15px*; padding:*5px*; font-weight:*bold*;">Status </td></tr>  <%  Connection con=**new** DBConnection().Connect();  PreparedStatement ps=con.prepareStatement("select \* from tbl\_enquiry"); ResultSet rs=ps.executeQuery();  **while**(rs.next()){  %>  <td style="padding:*5px*;"><%out.print(rs.getInt(1)); %></td>  <td style='padding:*5px*;'><%out.print(rs.getInt(2)); %></td>  <td style='padding:*5px*;'><%out.print(rs.getString(3)); %></td>  <td style='padding:*5px*;'><%out.print(rs.getString(4)); %></td>  <td style='padding:*5px*;'><%out.print(rs.getString(5)); %></td>  <td style='padding:*5px*;'><%out.print(rs.getString(6)); %></td>  <td style='padding:*5px*;'><%out.print(rs.getInt(7)); %></td>  <td style='padding:*5px*;'><%out.print(rs.getInt(8)); %></td>  <td style='padding:*5px*;'><%out.print(rs.getInt(9)); %></td>  <td style='padding:*5px*;'><a href=*'chstatus.jsp?enq\_id=*<%out.print(rs.getInt(1));  %>*'*><%out.print(rs.getString(11)); %></a></td>  </tr>  <%  //response.sendRedirect("viewinquiry.jsp");  }  //response.sendRedirect("viewinquiry.jsp");  %>  </table>  </td></tr></table>  </form>  </div>  <!-- /.container-fluid --> |

## Comments and Description

Comments are text notes added to the program to provide explanatory information about the source code. They are used in a programming language to document the program and remind programmers of what tricky things they just did with the code and also helps the later generation for understanding and maintenance of code. The compiler considers these as non-executable statements. Different programming language uses a different way of including the comments in the source code.

/\* this is a comment in JAVA.

This comment syntax is guaranteed to work on every compiler \*/

// This is also a comment in JAVA

// but it might present portability challenges

<!-- This is a comment in HTML-->

## Comments can be used for the various purpose like :

1. **Planning and reviewing :** In comments, we can write the pseudocode which we planned before writing the source code. Pseudocode is a mixture of natural language and high-level programming language. This helps in reviewing the source code more easily because pseudocode is more understandable than the program.

Example –

// GCD BY EUCLID'S ALGORITHM

/\* Euclid(m, n)

{

while m does not divide n r = n mod m

n = m m = r end

return m

}

\*/

In the above example, I have written the pseudo code of gcd in comments. This will help during the implementation of the code.

1. **Code description :** Code description is used by the programmer to make others understand his/her intent. It contains the summary of the code.

{

/\* A Multiline Comment -- Define a variable of string type and assign value to it\*/

string msg = "Haripada";

}

1. **Algorithmic description :** Comments are used for explanation of the methodology. Such explanations may include diagrams and formal mathematical proofs. This may constitute the explanation of the code, rather than a clarification of its intent. For example, a programmer may add a comment to explain why an insertion sort was chosen instead of a quicksort, as the former is, in theory, slower than the latter.

list = [ f(b), f(b), f(c), f(d), f(a), ... ];

// Need a stable sort. Besides,

// the performance really does not matter. insertion\_sort(list);

1. **Resource inclusion :** Logos, diagrams, and flowcharts consisting of ASCII art constructions can be inserted into source code formatted as a comment. Further, copyright notices can be embedded within source code as comments.
2. **Metadata :** Comments also contain metadata of the program. This metadata helps in software maintenance. The metadata includes the name of the creator of the original version, current maintainer of the program, data when the first version was created, the name of the people who have edited the program files so far etc.
3. **Debugging :** Brute force method is a common method of debugging. In this approach, print statements are inserted throughout the program to print the intermediate values with the hope that some of the printed values will help to identify the errors. After doing debugging we comment those print statements. Hence comment is also used for debugging.

int fun(int m)

{

int count = 0; while (m > 10) {

// printf("m is less than 10, m=%d", m); count++;

}

return m;

}

1. **Automatic documentation generation :** Programming tools sometimes store documentation and metadata in comments. These may include insert positions for automatic header file inclusion, commands to set the file’s syntax highlighting mode, or the file’s revision number. These functional control comments are also commonly referred to as annotations. Keeping documentation within source code comments is considered as one way to simplify the documentation process, as well as increase the chances that the documentation will be kept up to date with changes in the code.
2. **Stress relief :** Commenting on development tools, competitors, employers, working conditions, or the quality of the code itself are the ways to relieve stress. The occurrence of this phenomenon can be easily seen from online resources that track profanity in source code.

## Standardization of the coding /Code Efficiency

Coding standard are set up guidelines,best practices, programming style and conventions that developers adhere to when writing source code for a project. Here are the few guiddlines for the java coding standard.

**Coding Standards for Components:** It is recommended to write components name by its purpose. This approach improves the readability and maintainability of code.

**Coding Standards for Classes:** Usually class name should be noun starting with uppercase letter. If it contains multiple word than every inner word should start with uppercase.

Eg: String, StringBuffer, Dog

**Coding Standards for Interface:** Usually interface name should be adjective starting with uppercase letter. If it contains multiple word than every inner word should start with uppercase. Eg: Runnable, Serializable, Comparable

**Coding Standards for Methods:** Usually method name should either be verb or verb noun combination starting with lower letter. If it contains multiple word than every inner word should start with uppercase.

Eg: print(), sleep(), setSalary()

**Coding Standards for Variables:** Usually variable name should be noun starting with lowercase letter. If it contains multiple word than every inner word should start with uppercase.

Eg: name, age. mobileNumber

**Coding Standards for Constants:** Usually constant name should be noun. It should contain only uppercase If it contains multiple word than words are separated with ( \_ ) underscore symbol.

Usually we declare constants with public static and final modifiers.

**Java Bean Coding Standards:** A Java Bean is a simple java class with private properties and public getter and setter methods

## Getter Methods:

* It should be public method
* Method name should be prefixed with “get”
* It should not take any argument

## Setter Methods:

* It should be public method
* Return Type should be void
* Method name should be prefixed with “set”
* It should take some argument public class StudentBean{ private String name;

public void setName(String name){

this.name=name;

}

public String getName(){ return name;

}

}

Note: For boolean properties getter method can be prefixed with “get” or “is”

## Coding convention for Listners:

1. To register a Listner method name should prefixed with add Eg: public void addMyAccountListner( MyActionListner);

1. To unregister a Listner method name should prefixed with remove

Eg: public void removeMyAccountListner( MyActionListner);

## Error handling

Error are the most important part of any project which need special attention and requires proper error handling mechanism to make the software project more reliable . In java error can be classified in two types

1. Error types
2. Exception type

Errors are beyond the control of programmer which are handled by system itself but exception are the simple error which can be handle by the programmer. Java provides special mechanism for handling exception.

In this project all the source codes implements exception handling principle for handling all possible exception.

We follow following guidelines of exception handling which are as follows

1. Clean up Resources in a Finally Block or Use a Try-With-Resource Statement

It happens quite often that you use a resource in your try block, like an [*InputStream*](https://docs.oracle.com/javase/8/docs/api/java/io/InputStream.html), which you need to close afterward. A common mistake in these situations is to close the resource at the end of the try block.

public void doNotCloseResourceInTry() { FileInputStream inputStream = null; try {

File file = new File("./tmp.txt"); inputStream = new FileInputStream(file);

// use the inputStream to read a file

// do NOT do this inputStream.close();

} catch (FileNotFoundException e) { log.error(e);

} catch (IOException e) {

log.error(e);

}

}

The problem is that this approach seems to work perfectly fine as long as no exception gets thrown.

All statements within the try block will get executed, and the resource gets closed.

But you added the try block for a reason. You call one or more methods which might throw an exception, or maybe you throw the exception yourself. That means you might not reach the end of the try block. And as a result, you will not close the resources.

You should, therefore, put all your clean up code into the finally block or use a try-with-resource statement.

Use a Finally Block

In contrast to the last few lines of your try block, the finally block gets always executed. That happens either after the successful execution of the try block or after you handled an exception in a catch block. Due to this, you can be sure that you clean up all the opened resources.

public void closeResourceInFinally() { FileInputStream inputStream = null; try {

File file = new File("./tmp.txt"); inputStream = new FileInputStream(file);

// use the inputStream to read a file

} catch (FileNotFoundException e) { log.error(e);

} finally {

if (inputStream != null) { try {

inputStream.close();

} catch (IOException e) {

log.error(e);

}

}

}

}

Java 7’s Try-With-Resource Statement

Another option is the try-with-resource statement which I explained in more detail in my [introduction](https://stackify.com/specify-handle-exceptions-java/#tryWithResource) [to Java exception handling.](https://stackify.com/specify-handle-exceptions-java/#tryWithResource)

You can use it if your resource implements the [*AutoCloseable*](https://docs.oracle.com/javase/8/docs/api/java/lang/AutoCloseable.html) interface. That’s what most Java standard resources do. When you open the resource in the *try* clause, it will get automatically closed after the *try* block got executed, or an exception handled.

public void automaticallyCloseResource() { File file = new File("./tmp.txt");

try (FileInputStream inputStream = new FileInputStream(file);) {

// use the inputStream to read a file

} catch (FileNotFoundException e) { log.error(e);

} catch (IOException e) {

log.error(e);

}

}

1. Prefer Specific Exceptions

The more specific the exception is that you throw, the better. Always keep in mind that a co-worker who doesn’t know your code, or maybe you in a few months, need to call your method and handle the exception.

Therefore make sure to provide them as many information as possible. That makes your API easier to understand. And as a result, the caller of your method will be able to handle the exception better or [avoid it with an additional check.](https://stackify.com/top-java-software-errors/)

So, always try to find the class that fits best to your exceptional event, e.g. throw a [*NumberFormatException*](https://docs.oracle.com/javase/8/docs/api/java/lang/NumberFormatException.html) instead of an [*IllegalArgumentException*.](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html) And avoid throwing an unspecific *Exception*.

public void doNotDoThis() throws Exception { ... }

public void doThis() throws NumberFormatException { ... }

1. Document the Exceptions You Specify

Whenever you [specify an exception](https://stackify.com/specify-handle-exceptions-java/#specify) in your method signature, you should also [document it in your](http://blog.joda.org/2012/11/javadoc-coding-standards.html) [Javadoc](http://blog.joda.org/2012/11/javadoc-coding-standards.html). That has the same goal as the previous best practice: Provide the caller as many information as possible so that he can avoid or handle the exception.

So, make sure to add a *@throws* declaration to your Javadoc and to describe the situations that can cause the exception.

/\*\*

* This method does something extremely useful ...

\*

* @param input
* @throws MyBusinessException if ... happens

\*/

public void doSomething(String input) throws MyBusinessException { ... }

1. Throw Exceptions With Descriptive Messages

The idea behind this best practice is similar to the two previous ones. But this time, you don’t provide the information to the caller of your method. The exception’s message gets read by everyone who has to understand what had happened when the exception was reported in the log file or [your monitoring](https://stackify.com/error-monitoring/) [tool](https://stackify.com/error-monitoring/).

It should, therefore, describe the problem as precisely as possible and provide the most relevant information to understand the exceptional event.

Don’t get me wrong; you shouldn’t write a paragraph of text. But you should explain the reason for the exception in 1-2 short sentences. That helps your operations team to understand the severity of the problem, and it also makes it easier for you to analyze any service incidents.

If you throw a specific exception, its class name will most likely already describe the kind of error. So, you don’t need to provide a lot of additional information. A good example for that is the *NumberFormatException*. It gets thrown by the constructor of the class *java.lang.Long* when you provide a *String* in a wrong format.

try {

new Long("xyz");

} catch (NumberFormatException e) { log.error(e);

}

The name of the *NumberFormatException* class already tells you the kind of problem. Its message

only needs to provide the input string that caused the problem. If the name of the exception class isn’t that expressive, you need to provide the required information in the message.

17:17:26,386 ERROR TestExceptionHandling:52 - java.lang.NumberFormatException: For input stri ng: "xyz"

1. Catch the most specific exception first

Most IDEs help you with this best practice. They report an unreachable code block when you try to catch the less specific exception first.

The problem is that only the first catch block that matches the exception gets executed. So, if you catch an *IllegalArgumentException* first, you will never reach the catch block that should handle the more specific *NumberFormatException* because it’s a subclass of the *IllegalArgumentException*.

Always catch the most specific exception class first and add the less specific catch blocks to the end of your list.

You can see an example of such a try-catch statement in the following code snippet. The first catch block handles all *NumberFormatException*s and the second one all *IllegalArgumentException*s which are not a *NumberFormatException*.

public void catchMostSpecificExceptionFirst() {

try {

doSomething("A message");

} catch (NumberFormatException e) { log.error(e);

} catch (IllegalArgumentException e) { log.error(e)

}

}

1. Don’t catch Throwable

[*Throwable*](https://docs.oracle.com/javase/8/docs/api/java/lang/Throwable.html) is the superclass of all exceptions and errors. You can use it in a catch clause, but you should never do it!

If you use *Throwable* in a catch clause, it will not only catch all exceptions; it will also catch all errors. Errors are thrown by the JVM to indicate serious problems that are not intended to be handled by an application. Typical examples for that are the [*OutOfMemoryError*](https://docs.oracle.com/javase/8/docs/api/java/lang/OutOfMemoryError.html) or the [*StackOverflowError*](https://docs.oracle.com/javase/8/docs/api/java/lang/StackOverflowError.html). Both are caused by situations that are outside of the control of the application and can’t be handled.

So, better don’t catch a *Throwable* unless you’re absolutely sure that you’re in an exceptional situation in which you’re able or required to handle an error.

public void doNotCatchThrowable() { try {

// do something

} catch (Throwable t) {

// don't do this!

}

}

1. Don’t ignore exceptions

Have you ever analyzed a bug report where only the first part of your use case got executed?

That’s often caused by an ignored exception. The developer was probably pretty sure that it would never be thrown and added a catch block that doesn’t handle or logs it. And when you find this block, you most likely even find one of the famous “This will never happen” comments.

public void doNotIgnoreExceptions() { try {

// do something

} catch (NumberFormatException e) {

// this will never happen

}

}

Well, you might be analyzing a problem in which the impossible happened.

So, please, never ignore an exception. You don’t know how the code will change in the future. Someone might remove the validation that prevented the exceptional event without recognizing that this creates a problem. Or the code that throws the exception gets changed and now throws multiple exceptions of the same class, and the calling code doesn’t prevent all of them.

You should at least write a log message telling everyone that the unthinkable just had happened and that someone needs to check it.

public void logAnException() {

try {

// do something

} catch (NumberFormatException e) { log.error("This should never happen: " + e);

}

}

1. Don’t log and throw

That is probably the most often ignored best practice in this list. You can find lots of code snippets and even libraries in which an exception gets caught, logged and rethrown.

try {

new Long("xyz");

} catch (NumberFormatException e) { log.error(e);

throw e;

}

It might feel intuitive to log an exception when it occurred and then rethrow it so that the caller can

handle it appropriately. But it will write multiple error messages for the same exception.

17:44:28,945 ERROR TestExceptionHandling:65 - java.lang.NumberFormatException: For input stri ng: "xyz"

Exception in thread "main" java.lang.NumberFormatException: For input string: "xyz"

at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65) at java.lang.Long.parseLong(Long.java:589)

at java.lang.Long.(Long.java:965)

at com.stackify.example.TestExceptionHandling.logAndThrowException(TestExceptionHa ndling.java:63)

at com.stackify.example.TestExceptionHandling.main(TestExceptionHandling.java:58)

The additional messages also don’t add any information. As explained in best practice #4, the

exception message should describe the exceptional event. And the stack trace tells you in which class, method, and line the exception was thrown.

If you need to add additional information, you should catch the exception and wrap it in a [custom one](https://stackify.com/java-custom-exceptions/). But make sure to follow best practice number 9.

public void wrapException(String input) throws MyBusinessException { try {

// do something

} catch (NumberFormatException e) {

throw new MyBusinessException("A message that describes the error.", e);

}

}

So, only catch an exception if you want to handle it. Otherwise, specify it in the method signature and

let the caller take care of it.

1. Wrap the Exception Without Consuming it

It’s sometimes better to catch a standard exception and to wrap it into a [custom one](https://stackify.com/java-custom-exceptions/). A typical example for such an exception is an application or framework specific business exception. That allows you to add additional information and you can also implement a special handling for your exception class.

When you do that, make sure to set the original exception as the cause. The *Exception*class provides specific constructor methods that accept a *Throwable* as a parameter. Otherwise, you lose the stack trace and message of the original exception which will make it difficult to analyze the exceptional event that caused your exception.

public void wrapException(String input) throws MyBusinessException { try {

// do something

} catch (NumberFormatException e) {

throw new MyBusinessException("A message that describes the error.", e);

}

}

## CHAPTER-5

**TESTING**

## Introduction

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

## Testing techniques and testing strategies used along with the test case Designs and test reports.

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive at system testing, where the software and other system elements are tested as a whole.

UNIT TESTING

MODULE TESTING

# Component Testing

Integration Testing

SUB-SYSTEM TESING

SYSTEM TESTING

ACCEPTANCE TESTING

# User Testing

## 5.1.2. Unit Testing

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

## 1. WHITE BOX TESTING

This type of testing ensures that

1. All independent paths have been exercised at least once
2. All logical decisions have been exercised on their true and false sides
3. All loops are executed at their boundaries and within their operational bounds
4. All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form .we have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

## BASIC PATH TESTING

Established technique of flow graph with Cyclomatic complexity was used to derive test cases for all the functions. The main steps in deriving test cases were:

Use the design of the code and draw correspondent flow graph.

Determine the Cyclomatic complexity of resultant flow graph, using formula: V (G) =E-N+2 or

V (G) =P+1 or

V (G) =Number Of Regions

Where V (G) is Cyclomatic complexity, E is the number of edges,

N is the number of flow graph nodes, P is the number of predicate nodes.

Determine the basis of set of linearly independent paths.

## CONDITIONAL TESTING

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generate on particular condition is traced to uncover any possible errors.

## DATA FLOW TESTING

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable were declared. The *definition- use chain* method was used in this type of testing. These were particularly useful in nested statements.

## LOOP TESTING

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

1. All the loops were tested at their limits, just above them and just below them.
2. All the loops were skipped at least once.
3. For nested loops test the inner most loop first and then work outwards.
4. For concatenated loops the values of dependent loops were set with the help of connected loop.
5. Unstructured loops were resolved into nested loops or concatenated loops and tested as above.

Each unit has been separately tested by the development team itself and all the input have been validated.

## 5.1.3 Test Case:

The purpose of this document is to dictate which regression tests to run for the Online Inventory Management system when parts of the system are changed. They are a subset of the Test Plan & Cases for the system. Changes to the system may inadvertently cause changes to tests that have already passed; this document ensures that all potential affected test cases have been retested..

The scope is for development until the final delivery but also included is a potential scenario during which system administrator may upgrade the underlying infrastructure.

The focus of these tests is to ensure readiness for transition from development to production and confirms changes to the system were retested.

## Category View changed(Add,Update,Delete)

Tests to run after changes to Category View Page changes

|  |  |  |
| --- | --- | --- |
| **Test ID** | **Description** | **Rationale** |
| TC-08-01 | Add Category | May directly prevent Categoryr from  being added |
| TC-08-02 | Edit Category details | May directly prevent edit from being  edited. |
| TC-08-03 | Edit Category details  failed | May be unable to see details of Category. |
| TC-08-04 | Delete Category details | May directly prevent delete from being edited. |
| TC-09-01 | Check Category in table | May be unable see to add Category if  add done incorrectly |
| TC-10-01 | Update Category details fail | May be unable to update Category details if update done incorrectly. |
| TC-10-02 | View Category fails | May be unable to view Category details  when add done correctly. |
| TC-12-01 | Print Category details fail | May cause the generation of the report to fail if data became corrupt due to failure  to add or edit correctly |

## Package View Page changed (Add, Update and Delete)

Tests to run after changes to Package View Page changes

|  |  |  |
| --- | --- | --- |
| **Test ID** | **Description** | **Rationale** |
| TC-08-01 | Add Package | May directly prevent Package from being  added |
| TC-08-02 | Edit Package details | May directly prevent edit from being  edited. |

|  |  |  |
| --- | --- | --- |
| TC-08-03 | Edit Package y details  failed | May be unable to see details of Package. |
| TC-08-04 | Delete Package details | May directly prevent delete from being  edited. |
| TC-09-01 | Check Package in table | May be unable see to add Package if add done incorrectly |
| TC-10-01 | Update Package details  fail | May be unable to update Package details  if update done incorrectly. |
| TC-10-02 | View Package fails | May be unable to view Package details when add done correctly. |
| TC-12-01 | Print Package details fail | May cause the generation of the report to fail if data became corrupt due to failure  to add or edit correctly |

## Sub Category View changed (Add, Update and Delete)

Tests to run after changes to Sub Category View changes

|  |  |  |
| --- | --- | --- |
| **Test ID** | **Description** | **Rationale** |
| TC-08-01 | Add Sub Category | May directly prevent Sub Category from  being added |
| TC-08-02 | Edit Sub Category  details | May directly prevent edit from being  edited. |
| TC-08-03 | Edit Sub Category  details failed | May be unable to see details of Sub  Category . |
| TC-08-04 | Delete Sub Category  details | May directly prevent delete from being  edited. |
| TC-09-01 | Check Sub Category in  table | May be unable see to add Sub Category  if add done incorrectly |
| TC-10-01 | Update Sub Category  details fail | May be unable to update Sub Category  details if update done incorrectly. |
| TC-10-02 | View Sub Category fails | May be unable to view Sub Category details when add done correctly. |
| TC-12-01 | Print Sub Category details fail | May cause the generation of the report to  fail if data became corrupt due to failure to add or edit correctly |

## User View changed (Add, Delete and Update)

Tests to run after changes to Education View Page changes

|  |  |  |
| --- | --- | --- |
| **Test ID** | **Description** | **Rationale** |
| TC-08-01 | Add User | May directly prevent User from being  added |
| TC-08-02 | Edit User details | May directly prevent edit from being  edited. |
| TC-08-03 | Edit User details failed | May be unable to see details of User. |
| TC-08-04 | Delete User details | May directly prevent delete from being  edited. |
| TC-09-01 | Check User in view  table | May be unable see to add User if add  done incorrectly |

|  |  |  |
| --- | --- | --- |
| TC-10-01 | Update User details fail | May be unable to update User details if  update done incorrectly. |
| TC-10-02 | View User fails | May be unable to view User details when add done correctly. |
| TC-12-01 | Print User details fail | May cause the generation of the report to fail if data became corrupt due to failure  to add or edit correctly |

## Report Generation changed

Tests to run after changes to Report Page changes

|  |  |  |
| --- | --- | --- |
| **Test ID** | **Description** | **Rationale** |
| TC-08-01 | Print Report | May directly prevent printing report. |
| TC-08-02 | Printing format error | May directly prevent printing format  error if prescribed paper given to printer. |
| TC-08-03 | Total amount of page is  not visible | May be unable to see Total amount of in  printing report. |

1. Login Authorization changes

Tests to run after updates to any login authorization change.

|  |  |  |
| --- | --- | --- |
| **Test ID** | **Description** | **Rationale** |
| TC-01-01 | Valid user login | May directly prevent  logins to the system |
| TC-02-01 – TC-02-  02 | Administrator logins | May indirectly invalidate  role-based login |
| TC-03-01 | Create new user | May indirectly prevent the new logins from being stored in the database May indirectly allow  unauthorized access to add or edit User |
| TC-04-01 | Add user timecard | May allow unauthorized access to add an user  timecard |
| TC-05-01 | Edit user timecard | May allow unauthorized access to edit an user  timecard |
| TC-09-01 – TC-09- 02 | Check in item | May allow unauthorized access to add or edit product transactions to  non-administrators |

*1. MySQL / Apache Tomcat Server upgrade*

Regression tests when any part of the infrastructure is upgraded

|  |  |  |
| --- | --- | --- |
| **Test ID** | **Description** | **Rationale** |
| TC-01 – TC-12 | All test cases | Database, web server  upgrade may indirectly |

|  |  |  |
| --- | --- | --- |
|  |  | cause any of the templates  to fail. |
| TC-13-01 | Data loss of Make My Tour | Database upgrade may  indirectly cause data loss. |
| Test Case Number | **TC-13-01** | |
| Test Item | **Make My Tour** | |
| Test Priority | **Could have** | |
| Pre-conditions | **The system administrator has successfully upgrade one or more of the following (MySQL, ApacheTomcat) and knows the Sub Category list count and the product list**  **count** | |
| Post-conditions | **User has taken the new category count and the product**  **list count** | |
| Input Specifications | 1. **User logs in to main index page** 2. **User selects Admin login page** 3. **User goes to Category** 4. **User takes count of Category** | |
| Expected Output  Specifications | **The pre-upgrade counts match the post-upgrade counts** | |
| Pass/Fail Criteria | **Fail if loss of user data or product data is indicated by counts less than the original**  **Fail if duplication of data or entry of extraneous data is indicated by counts higher than the original** | |
| Assumptions and  Constraints | **Assume that the CRCD would upgrade sometime in the**  **future while still using the system.** | |
| Dependencies | **None** | |
| Completion Criteria | **All Sub Category data and product data counts match (data preserved).** | |
| Traceability | **N/A – Not a requirement but a feature provided by Haripada** | |

1. Regression Test Results

After the Core Capability Drive Through, enhancements requested by the user were made to the system and a two iteration regression test was executed.

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Iteration 1** | **Iteration 2** |
| TC-01-01 | Pass | Pass |
| TC-02-01 | Pass – but non-admin sees nulls for Make My Tour add page in date fields (admin acct works ok). | Pass – but non-admin sees nulls for category add page in date fields (admin acct  works ok). |
| TC-03-01 | Pass | Pass |
| TC-04-01 | Pass | Pass |

|  |  |  |
| --- | --- | --- |
| TC-04-03 | Pass | Pass |
| TC-05-01 | Pass | Pass |
| TC-07-01 | Pass | Pass |
| TC-08-01 | Pass | Pass |
| TC-09-01 | Pass | Pass |
| TC-10-01 | Pass | Pass |
| TC-11-02 | Fail : User entries were no longer integrated  (1st regression test iteration) | Pass |
| TC-12-01 | Pass | Pass |
| TC-13-01 | N/A – speculated test case | N/A : speculated test case |

## CHAPTER-6

**SYSTEM SECURITY MEASURES**

* 1. Introduction

The protection of computer based resources that includes hardware, software, data, procedures and people against unauthorized use or natural

Disaster is known as System Security.

System Security can be divided into four related issues:

1. Security
2. Integrity
3. Privacy
4. Confidentiality

**System Security** refers to the technical innovations and procedures applied to the hardware and operation systems to protect against deliberate or accidental damage from a defined threat.

**Data Security** is the protection of data from loss, disclosure, modification and destruction.

**System Integrity** refers to the power functioning of hardware and programs, appropriate physical security and safety against external threats such as eavesdropping and wiretapping.

**Privacy** defines the rights of the user or organizations to determine what information they are willing to share with or accept from others and how the organization can be protected against unwelcome, unfair or excessive dissemination of information about it.

**Confidentiality** is a special status given to sensitive information in a database to minimize the possible invasion of privacy. It is an attribute of information that characterizes its need for protection.

## 6.2 System Security Measures

In addition, utilizing website vulnerability scanners alone to protect your website does not guarantee security. Vulnerability scanners usually neglect the security of custom web applications that reside on the web server. This means that if a custom web application has a security flaw; it will not be detected since the scanner is programmed to identify flaws within its signature database.

In order to be truly secure, a web application should be immune to SQL injections, session hijackings, session tampering, cross site scripting and brute force attacks among other vulnerabilities. How can you go about ensuring your web application is truly secure?

Application access

Application access security is implemented by "Roles and Rights Management." For example, in our projects should not give User access to a to generate report since their job is only to sales the product and updating profile. Any menus, forms and screens related to the admin should therefore not be visible. A thorough testing of all roles and rights is thus needed to ensure that every role accesses its own screens, forms and modules.

Data protection

Data encryption is a critical security test. Any data stored in the database like passwords, user accounts and credit card information must always be encrypted. In addition, the flow of critical business information, even within the application must be encrypted to make it safe. When testing for data security, a tester should query the database for business critical and sensitive data to verify that it has been saved in an encrypted format. Of importance are the different "submit" actions and whether any information is being displayed in the address bar of the web browser.

SQL injections

SQL injections are the most prevalent and dangerous attacks on the Internet today. They take advantage of web application vulnerabilities to take control of databases which exposes confidential information. Auditing and remediation of exploitable software vulnerabilities should be an ongoing approach to identify and block any back-end security holes. Recently, Oracle released the Oracle Database Firewall that helps protect against SQL exploits. The firewall provides multiple layers of security making it difficult to penetrate protected databases.

XSS (Cross Site Scripting)

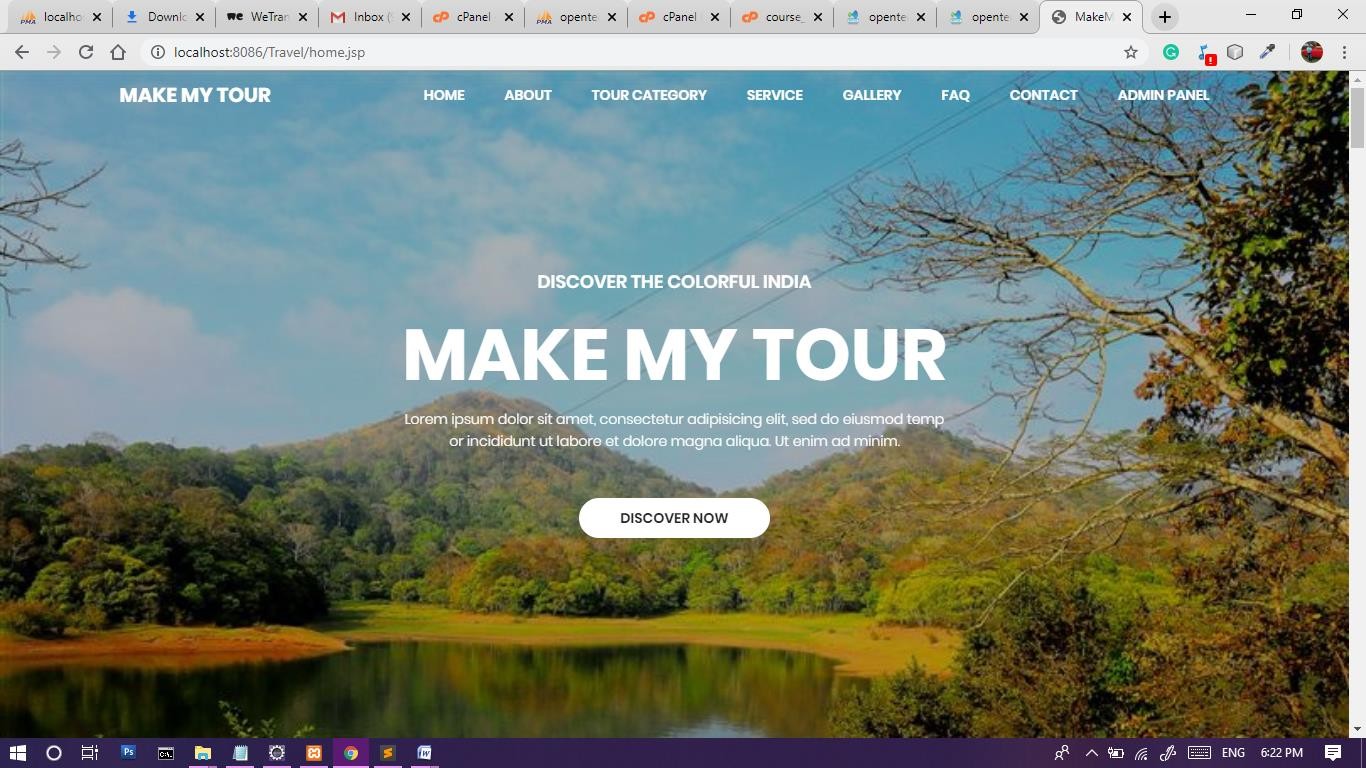
XSS vulnerabilities arise where web applications dynamically include data from users in web pages without proper validation. By use of JavaScript embedded on a web page, a malicious user can control the victim's browser bypassing normal security restrictions. To protect against XSS, developers must ensure that dynamically-generated pages do not contain undesired tags. Filtering is one approach that developers can use to ensure that all script tags are validated in the out process. Another approach would be to perform server-side encoding for all scripting tags to prevent unintended execution of scripts.

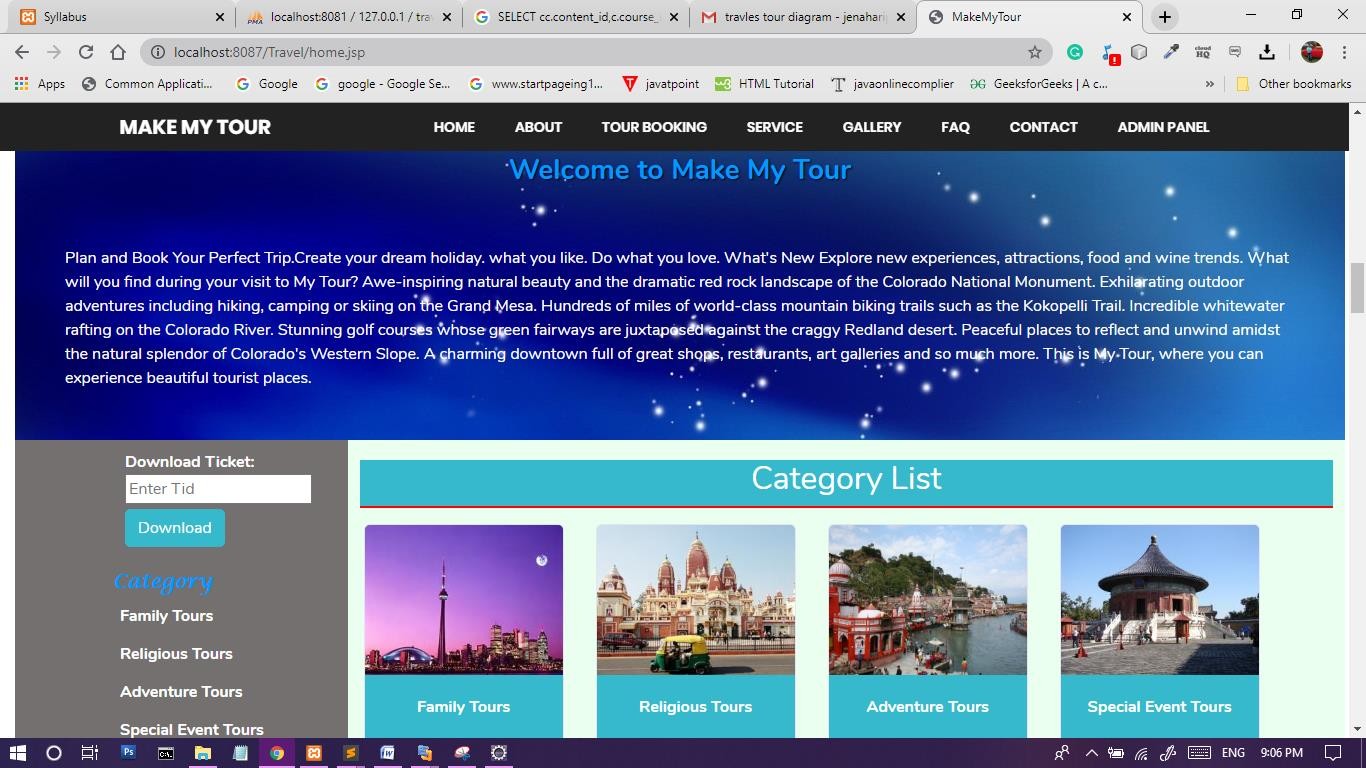
Other ways to ensure web application security includes brute-force attack prevention by account suspension or blocking when login attempts fail and the testing of service access points to ensure that data downloads and uploads have set security restrictions such as virus scanning and file size limits.

## CHAPTER-7

**SCREEN**

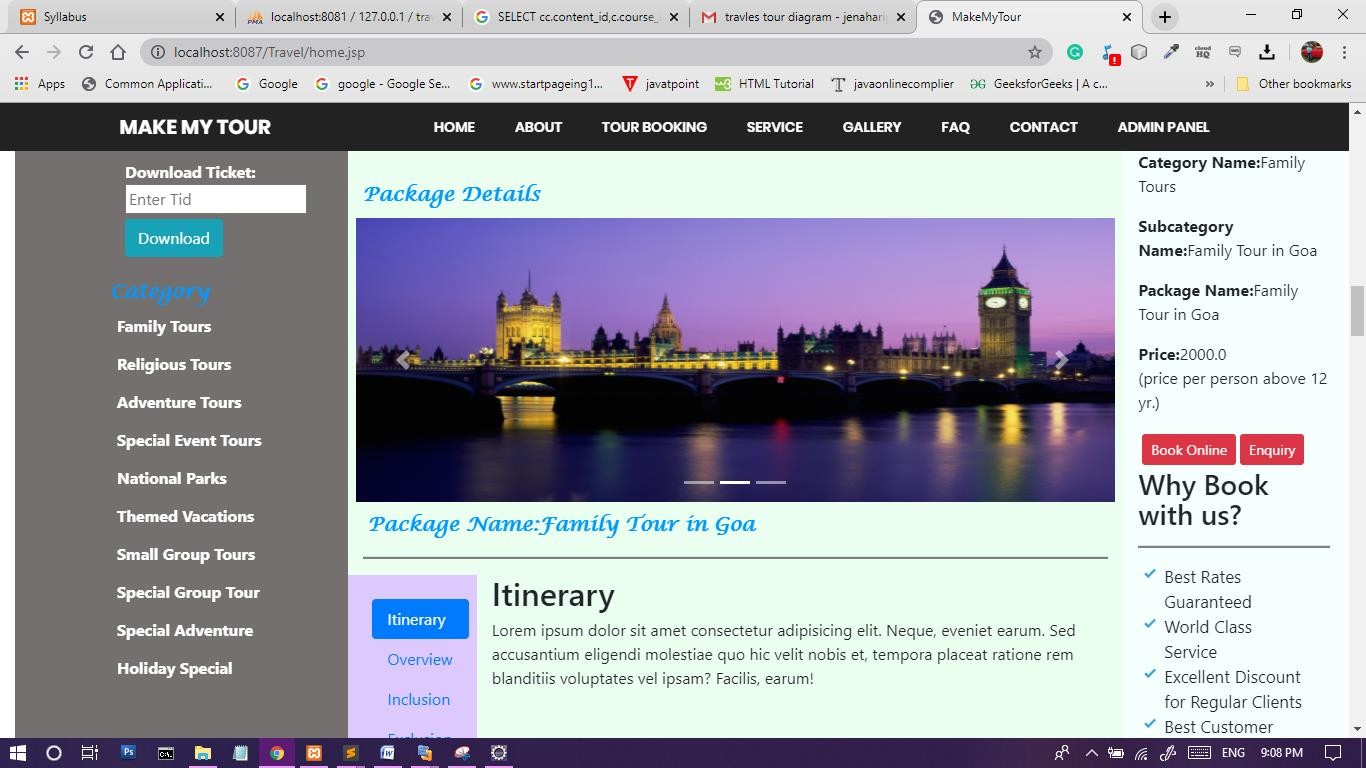
## Home page



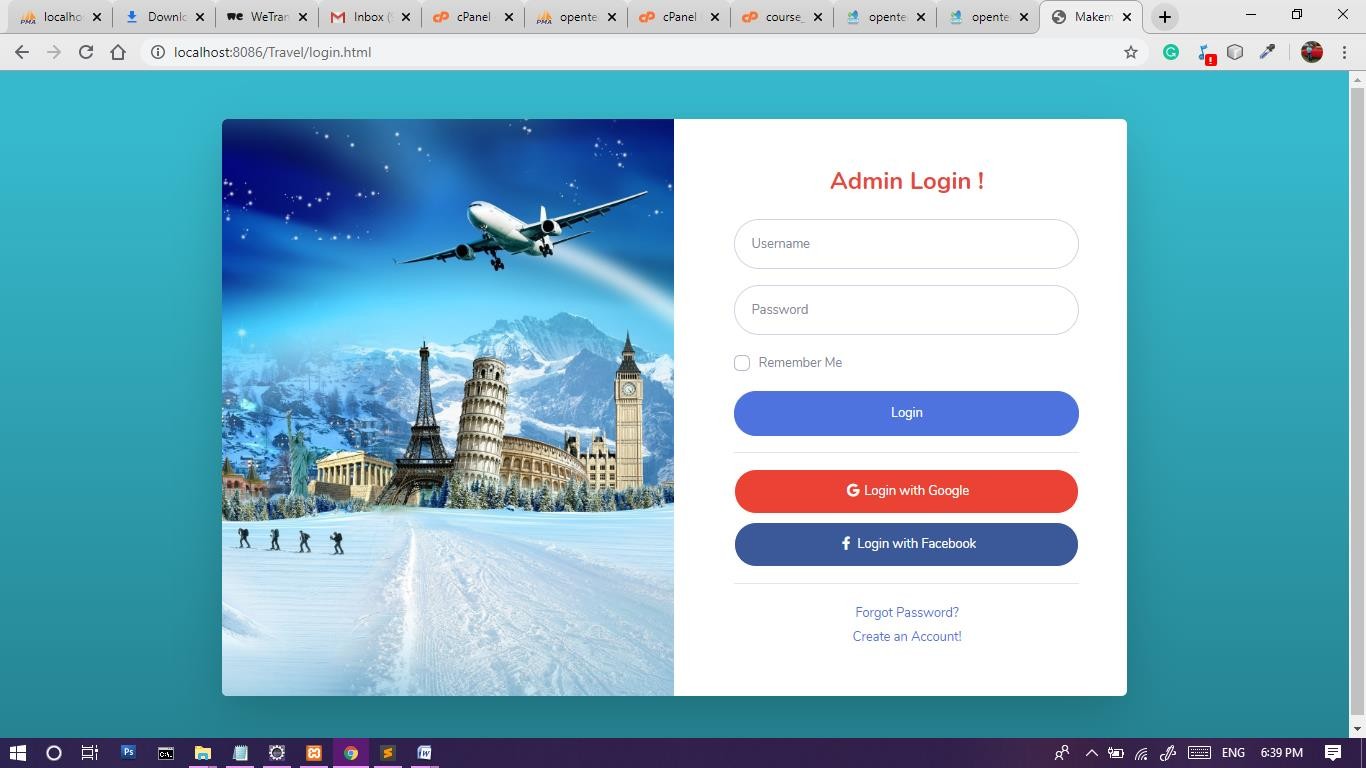
1. **Tour Category**

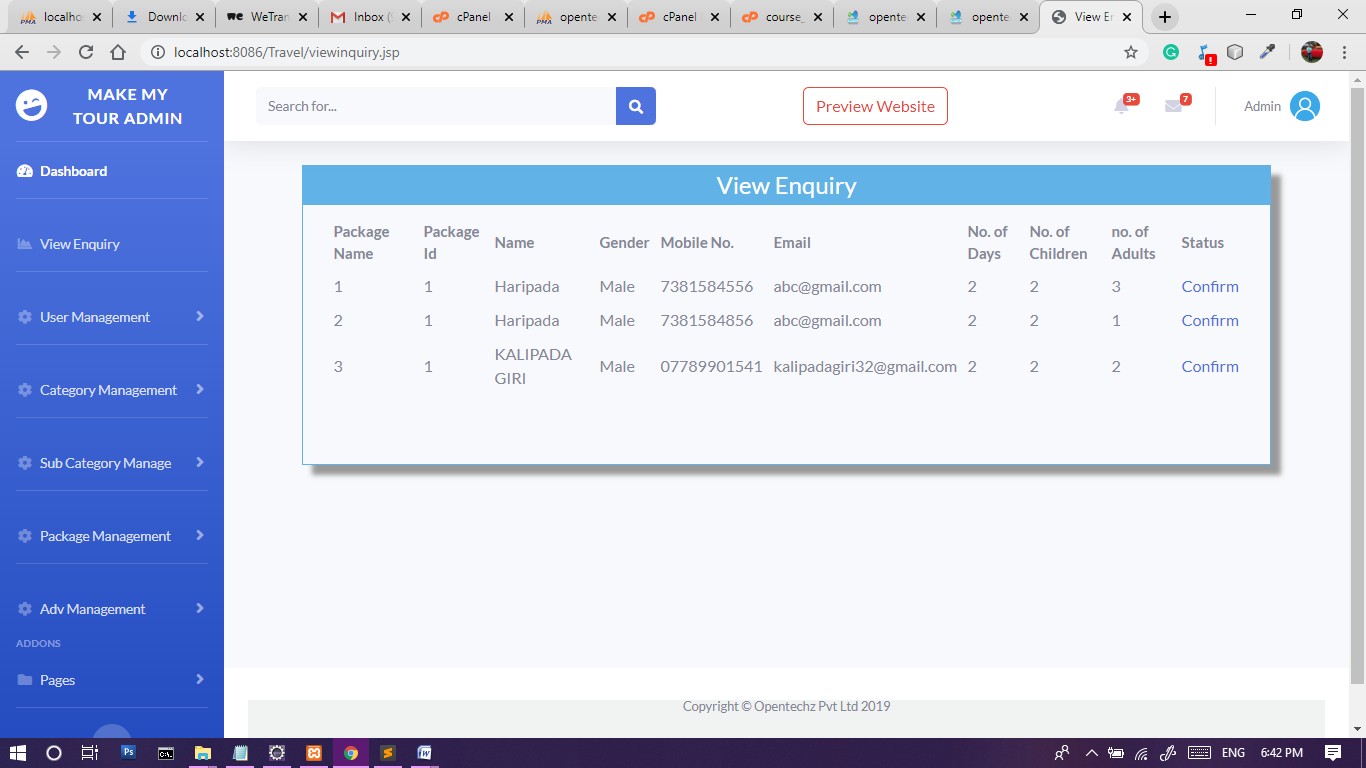
## Subcategory

1. **View Package**

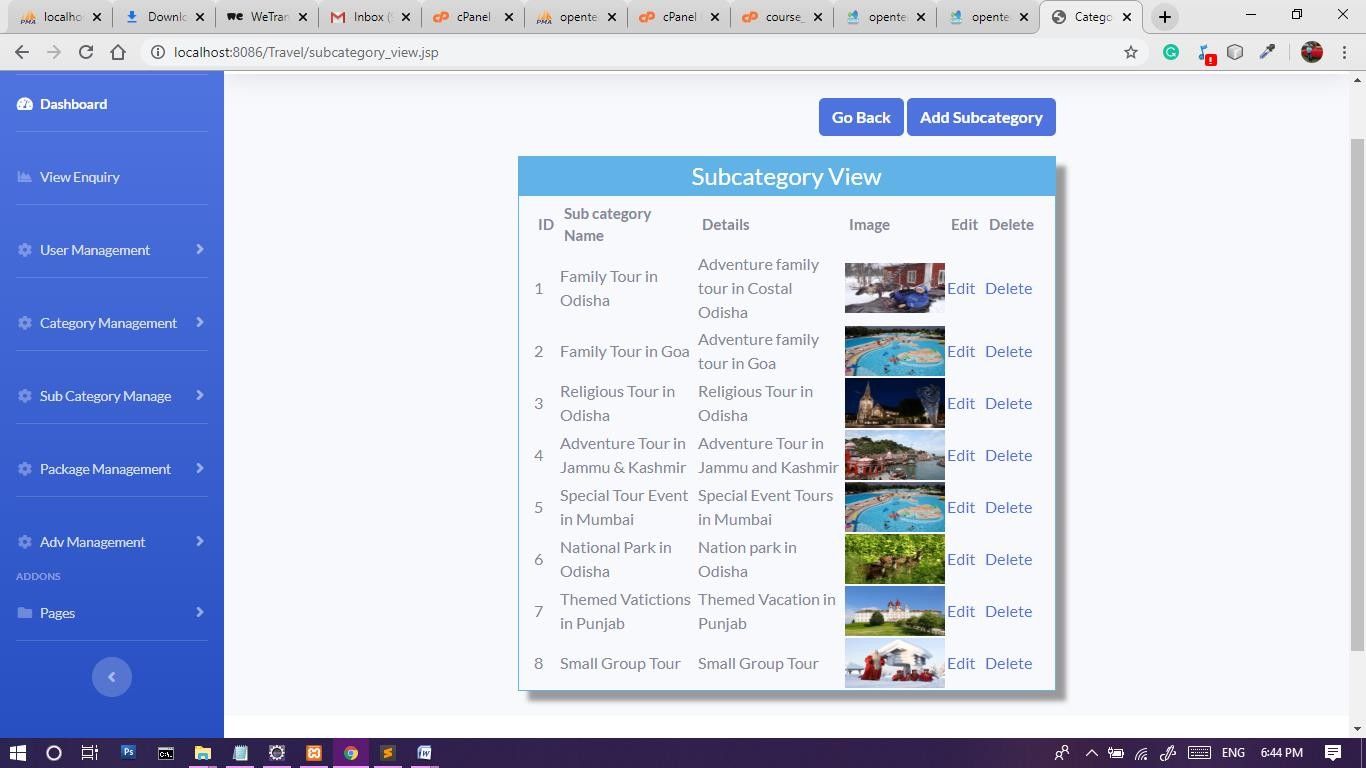


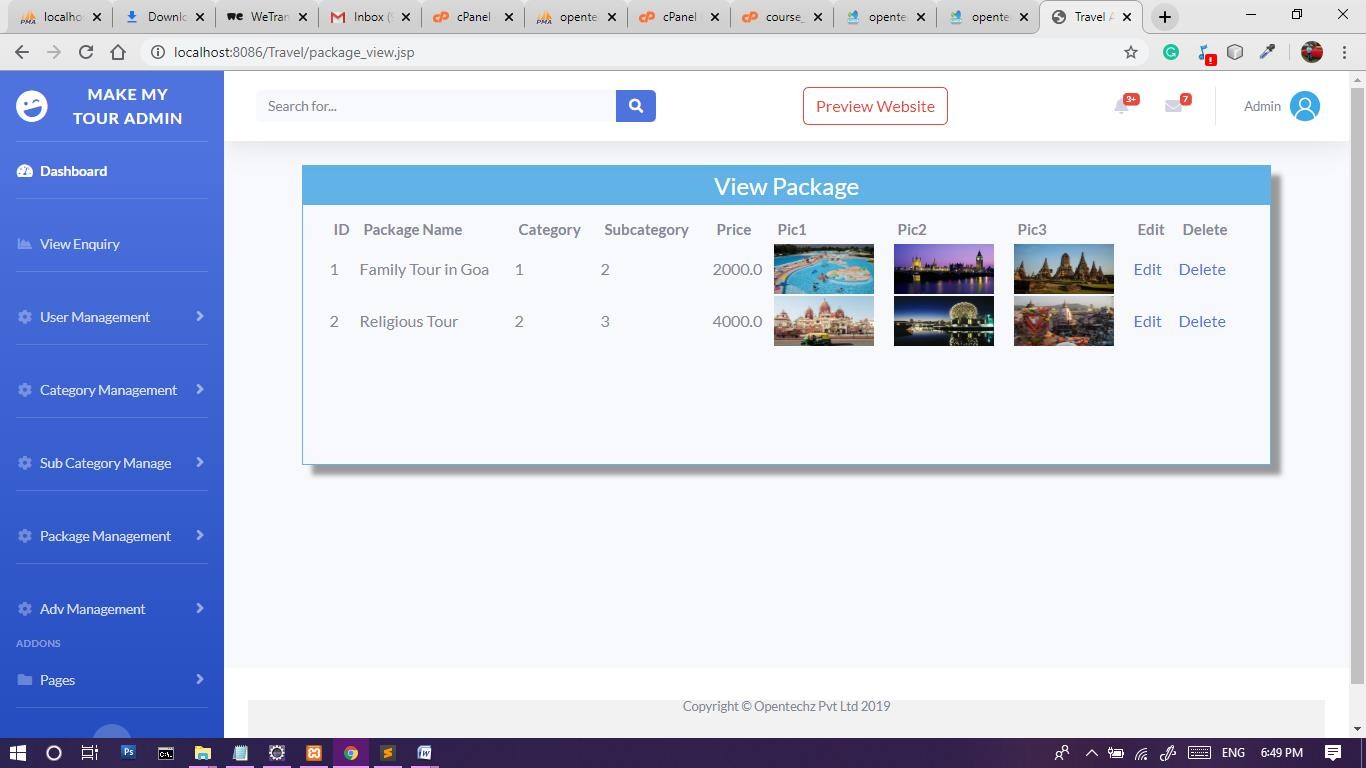
## Admin login



1. **Admin View Inquiry**

## Admin SubCategory Management



1. **Admin Package Management**

## CHAPTER-8

**FUTERE SCOPE & CONCLUSION**

## SCOPE FOR FURTHER DEVELOPMENT Future Enhancements:

It is not possible to develop a system that makes all the requirements of the user. User requirements keep changing as the system is being used. Some of the future enhancements that can be done to this system are:

1. As the technology emerges, it is possible to upgrade the system and can be adaptable to desired environment.
2. Because it is based on object-oriented design, any further changes can be easily adaptable.
3. Based on the future security issues, security can be improved using emerging technologies.
4. sub admin module can be added

## CONCLUSION

The **“Make My Tour”** was successfully designed and is tested for accuracy and quality. During this project we have accomplished all the objectives and this project meets the needs of the organization. The developed will be used in searching, retrieving and generating information for the concerned requests.

## GOALS

* + 1. Reduced entry work
    2. Making Tour booking is very easy and fast with effective manner.
    3. Most Secure web application.
    4. Easy retrieval of information
    5. Reduced errors due to human intervention
    6. User friendly screens to enter the data
    7. Portable and flexible for further enhancement
    8. Web enabled.
    9. Fast finding of information requested

## CHAPTER-9

**BIBLIOGRAPHY / REFERENCES**

## For Java installation

* 1. https:/[/www.java.com/](http://www.java.com/en/download/)e[n/download/](http://www.java.com/en/download/)

## For MySql Database installation

* 1. <https://dev.mysql.com/downloads/mysql/>

## For Server(Apache Tomcat) installation

* + 1. <https://tomcat.apache.org/download-90.cgi>

## Reference websites

* + 1. [https://www.javatpoint.com](https://www.javatpoint.com/)
    2. [https://www.bootstrap.com](https://www.bootstrap.com/)
    3. [https://www.w3schools.com](https://www.w3schools.com/)
    4. <http://www.tutorialspoint.com/java/index.htm>
    5. <https://en.wikipedia.org/wiki/Wiki>
    6. <https://www.geeksforgeeks.org/>

## Reference Books

* + 1. Complete reference Java
    2. Thinking in java
    3. Java Black Book J2EE
    4. Learn Java in Easy Steps
    5. Complete reference Java
    6. Object-Oriented Analysis &Design Using UML