SYSTEM FOR ADMINISTRATION OF CLIENT REQUESTS

By

Nitin V. Ramrakhiyani
04BIT036

at

Tata Consultancy Services Ltd.
Gandhinagar

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Ahmedabad 382481
April 2008

SYSTEM FOR ADMINISTRATION OF CLIENT REQUESTS

Major Project

Submitted in partial fulfillment of the requirements

For the degree of

Bachelors of Technology in Information Technology Engineering

By

Nitin V. Ramrakhiyani
04BIT036

Guided by

Prof. T. U. Upadhyay
Assistant Professor,
Department of CS Engineering,
Institute of Technology,
Nirma University,
Ahmedabad

Mr. Amit Maheshwari
Assistant Consultant,
Tata Consultancy Services,
Gandhinagar

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Ahmedabad 382481
April 2008

CERTIFICATE

This is to certify that the Major Project entitled "System For Administration of Client Requests" submitted by Mr. Nitin Vijaykumar Ramrakhiyani, towards the partial fulfillment of the requirements for the degree of Bachelors of Technology in Information Technology Engineering of Nirma University of Science and Technology, Ahmedabad is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. T. U. Upadhyay
Guide,
Assistant Professor,
Department of CE/IT Engineering,
Institute of Technology,
Nirma University,
Ahmedabad

Prof. D. J. Patel
Professor and Head,
CE_IT_MCA Department,
Institute of Technology,
Nirma University,
Ahmedabad

ACKNOWLEDGEMENT

"Gratitude is a feeling which is more eloquent than words, more tranquil than silence…"

It gives me great pleasure in expressing thanks and profound gratitude to **Mr. Amit Maheshwari,** Assistant Consultant, Tata Consultancy Services Limited for his continual support throughout the project Work. I would also like to thank my Faculty Guide, **Prof. T. U. Upadhyay**, Assistant Professor, Department of Computer Engineering, Institute of Technology, Nirma University, Ahmedabad for her valuable guidance and continual encouragement throughout the Major project.

I am also thankful to **Prof. D. J. Patel**, Head, Department of Computer Science And Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his continual kind words of encouragement and motivation throughout the Major Project. I am also thankful to **Prof. A. B. Patel**, Director, Institute of Technology for his kind support in all respects during my study.

I am thankful to all faculty members of Department of Information Technology Engineering, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

I am also thankful to the employees of TCS, working with us in the same project, for the kind support they gave us.

The blessings of God and my family members made the way for completion of the major project. I am very much grateful to them.

I am immensely thankful to my friends, who always stood beside and motivated me throughout this course.

Nitin Vijaykumar Ramrakhiyani
Roll No. 04BIT036

ABSTRACT

The project named "SYSTEM FOR ADMINISTRATION OF CLIENT REQUESTS" is being developed for a financial organization. This system acts an interface between the firm and its customers.

On the customer side, this system allows the users to register their requests, query those requests and to change the status of their requests. However, to an organization, the system would allow the administrator to set up a new consultant to care of the various tasks that are to be carried out in order to get a request satisfied. This system even allows the administrator to add new users to the consultant and to disable the users already assigned to a consultant. The system grants all the rights to the administrator to query a consultant, task, and to change their status.

The main objective of this system is to provide an all round support to the customers in terms of their request which can be as simple as – Changing the account type or as intricate as – Getting a loan. However, a request is a composition of a set of pre-requisite tasks which when fulfilled would allow further processing. These tasks may include - Document Gathering, Document Verification etc. These tasks are then taken up by a specific consultant which is an assembly of users. A consultant may accept or reject a task if the task does not fall in its domain. The system takes into account all the minuscule details before the dispensation of the request.

Even though the system is developed for a financial organization, it can be customized to suite the requirements of other institutes or organizations including educational institutes.

The system finds further scope in its parent module as an integral part. It actually carves a window for communication between the customers and the module and in turn the organization.
Preliminary work that was carried out involved construction of JSF based GUI and report designing using Crystal Report Designer. This preliminary work laid a basic foundation of the technologies to be used further on.

Various findings emerged such:

- Enormous Scope of Java.
- Scope of various kinds of Java and Java-based technologies.
- Widely used SDLC practices.
- Change management that takes place at various places.
- Various kinds of deliverables and delivery times during SDLC.
- Work culture and organizational behavior.

CONTENTS

Certificate I

Acknowledgement III

Abstract IV

Contents VI

List of figures IX

List of tables IX

List of Abbreviations and Nomenclature X

CHAPTER 1 INTRODUCTION

1.1 About the Company 1

1.1.1 Introduction of the Company 1

1.1.2 Quality Policy 5

1.1.3 Communication 6

1.2 The System 6

1.2.1 Definition of the System 6

1.2.2 Project Development Model 6

1.2.3 Concerned Audiences and Users 8

1.2.4 Purpose and Objectives 8

1.2.5 Proposed System 9

1.3 Project Profile 14

1.3.1 Project Title 14

1.3.2 Scope of the Project 14

1.3.3 Project Team 15

1.3.4 Hardware/Software Environment in the company 16

CHAPTER 2 SYSTEM ANALYSIS

2.1 Feasibility Study 17

2.1.1 Operational Feasibility 17

2.1.2 Technical Feasibility 18

2.1.3 Financial and Economic Feasibility 19

2.2 Requirement Analysis 19
2.2.1 Fact Finding Techniques 19

2.3 Context Diagram 23

2.4 Data Flow Diagrams 24
2.4.1 Level I Diagrams 24
2.4.2 Level II Diagrams 25

CHAPTER 3 SYSTEM DESIGN
3.1 System Architecture 29
3.1.1 Model View Controller Architecture 29
3.1.2 Project Architecture (Detailed) 32

3.2 Class Diagram 35

3.3 Sequence Diagrams (Flow) 37

3.4 Data Dictionary and Table Diagram 39

CHAPTER 4 USER MANUAL
4.1 Screens/GUI (with Description) 47

CHAPTER 5 TESTING 54

CHAPTER 6 FUTURE ENHANCEMENT 58

CHAPTER 7 APPENDIX
7.1 Description of the Technology Used 60
7.1.1 Java Server Faces 60
7.1.2 Hibernate 64
7.1.3 Spring 68

7.2 Description of the Tools Used 71
7.2.1 IBM WebSphere Integration Developer 71
7.2.2 PL/SQL Developer 72
7.3.3 Crystal Report Designer 74
7.3.4 Sybase PowerDesigner 75
7.3.5 Oracle 10g 76

CHAPTER 8 SUMMARY AND CONCLUSION

8.1 Summary 78
8.2 Conclusions 78

CHAPTER 9 BIBLIOGRAPHY AND REFERENCES

9.1 Bibliography 79
9.2 References 79

List of Figures

Figure 1.1	Project Team Structure	15
Figure 2.1	Context Level Diagram	23
Figure 2.2	Level I DFD	24
Figure 2.3	Level II DFD – Search Proceeding	25
Figure 2.4	Level II DFD – Add Proceeding	26
Figure 2.5	Level II DFD – Modify Proceeding	27
Figure 2.6	Level II DFD – Disable Proceeding	28
Figure 3.1	MVC Architecture	30
Figure 3.2	Common MVC Architectural Flow	30
Figure 3.3	Project Architecture	32
Figure 3.4	Class Diagram	35
Figure 3.5	Sequence Diagram – Search and Disable Proceeding	37
Figure 3.6	Sequence Diagram – Add Proceeding	38
Figure 3.7	Sequence Diagram – Modify Proceeding	38
Figure 3.8	Table Diagram	46
Figure 4.1	Screenshot of Main Menu	47
Figure 4.2	Screenshot of Search Type of Proceeding	48
Figure 4.3	Screenshot of Search Type of Proceeding	49
Figure 4.4	Screenshot of Add Type of Proceeding	50
Figure 4.5	Screenshot of Add Type of Proceeding with error msg	51
Figure 4.6	Screenshot of Modify Type of Proceeding	52
Figure 4.7	Screenshot of Modify Type of Proceeding with msg	53
Figure 7.1	JSF Life Cycle	61
Figure 7.2	Work Flow (JSF Request)	63
Figure 7.3	Hibernate Architecture	65
Figure 7.4	Spring Architecture	69
Figure 7.5	Association Diagram	71
Figure 7.6	Report Generation Phases	74

List of Tables

Table 1.1	Hardware/Software Being Used
Table 3.1	List of Columns of the table T_TYPE_OF_PROCEEDING
Table 3.2	List of Columns of the table T_PROCEEDING_IN_BRANCH
Table 3.3	List of columns of the table T_PROCEEDING_IN_OPERATION
Table 3.4	List of columns of the table T_OPERATION_TASK
Table 3.5	List of columns of the table T_TYPE_OF_TASK
Table 3.6	List of columns of the table T_TYPE_OF_CONSULTANT
Table 3.7	Description of SEQ_KEY_TYPE_OF_PROCEEDING
Table 3.8	Description of SEQ_ID_TYPE_OF_ PROCEEDING
Table 5.1	UTC – Search Type of Proceeding
Table 5.2	UTC – Add Type of Proceeding
Table 5.3	UTC – Modify Type of Proceeding
Table 7.1	Features of Oracle 10g

List of Abbreviations Used

Abbreviations Used	Description
SACR	System for administration of client requests.
TOC	Type of Consultant.
TOT	Type of Task
TOP	Type of Proceeding.
SDD	Software Design Document.
UTC	Unit-Test cases.
BPEL	Business Process Execution Language.
WSDL	Web Service Definition Language.
DAO	Data Access Objects.
EJB	Enterprise Java Beans.
JNDI	Java Naming Directory Interface.
IOC	Inversion Of Control.
AOP	Aspect Oriented Programming.
POJO	Plain Old Java Objects.

Nomenclature Used

Nomenclature Followed	Description
Consultant	Department.
User	Personnel.
Proceeding	Request.
Branch	Type of Proceeding
Operation	Type of Operation.

1. INTRODUCTION

1.1 ABOUT THE COMPANY

1.1.1 Introduction of the Company

Tata Consultancy Services Limited (TCS Limited) is one of the world's largest providers of information technology, consulting, services and business-process outsourcing which commenced operations in 1968. Tata Consultancy Services (TCS) provides innovative, end-to-end consulting, IT solutions and services to Fortune 500 clients in over 50 countries. With over 1,10,000 consultants, over 100 branches, alliances with world IT leaders, cutting-edge R&D, and world-class infrastructure, TCS offers products and services in e-Business, Application Development, Architecture & Technology Consulting, Engineering Services, e-Security, Large Projects, Quality Consulting and Infrastructure Development, across verticals such as Banking, Financial, Services, Insurance, Telecom, Manufacturing, Retail & Distribution, Transportation, Life Sciences & Healthcare, Government and Utilities. The company generated consolidated revenues of US $4.3 billion for fiscal year ended 31 March 2007 and is listed on the National Stock Exchange and Bombay Stock Exchange in India.

Vision

To be Global Top 10 by 2010

Mission

To help Customers achieve their business objectives, by providing Innovative, Best-in-class Consulting, IT solutions and services. We shall make it a joy for all stakeholders to work with us.

Values

Integrity,
Leading change,
Excellence,
Respect for individual, and Learning & Sharing.

Business Units

The various business units of TCSL are:-

- Tata Research Development and Design Centre
- TCS Innovation Labs
- TCS Advanced Technology Centre (ATC)
- Centers of Excellence

National and International Presence

Currently TCS has branches in the following Indian cities: Hyderabad, Bhavnagar, Mumbai, Chennai, Kolkata, Delhi, Bangalore, Thiruvananthapuram, Lucknow, Ahmedabad, Pune, Nagpur, Jamshedpur, Gandhinagar, Coimbatore, Kochi, Gurgaon, NOIDA, Bhubaneswar, Nabadwip and Goa.

TCS has its international presence in the countries namely: South Africa, Morroco, China, HongKong, Indonesia, Japan, Malaysia, Saudi Arabia, Singapore, South Korea, Taiwan, UAE, Australia, Denmark, Finland, France, Germany, Hungary, Iceland, Ireland, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland, United Kingdom, Canada, Mexico, USA, Argentina, Brazil, Chile, Colombia, Ecuador, Uruguay.

National Acquisitions and Contributions

TCS launched a mission-mode e-governance project, MCA-21, for the Ministry of Company Affairs, Government of India, and has created an automated solution for the National Rural Employment Guarantee scheme that reduces waste and corruption.

TCS is a part of National Skills Registry (NSR), a centralized database of all employees of the IT services and BPO companies in India. The objective of NSR is to improve recruitment practices in IT and BPO industry, which will in turn help in maintaining India's global competitive advantage.

International Acquisitions and Contributions

The Company's Global Network Delivery Model and the "Experience Certainity" Slogan has given them unparalleled execution capabilities across the world and has greatly enhanced their ability to consult and execute multi-location, complex engagements. In the recent times TCS has been successful in aquiring many international firms like Chile's Comicrom S.A., Australia's Financial Network Services (Holdings) Pty Ltd, (FNS) and Sweden's Indian IT Resources.

The Company has set up two subsidiaries, viz. TCS FNS Pty Limited in Australia and Diligenta Limited in the UK. It has also set up other subsidiaries such as Portugal Unipessoal Limitada in Portugal, Tata Consultancy Services Luxembourg S.A. in Luxembourg and Tata Consultancy Service Chile Limited in Chile. The Company has 49 subsidiaries as on March 31, 2006.

In March 2006, the Company, through its subsidiary Diligenta Limited acquired, on a going concern basis certain businesses of Pearl Group Services Limited. The acquisition included specified insurance contracts and claim administration business and assets including goodwill and know-how.

The Company has entered into a joint venture agreement with the Intelenet Global Services Ltd. For the GM deal, TCS is tying up with EDS to bid for parts of the business.In February, 2007, TCS kicked off a joint venture in China with Microsoft and three Chinese entities.

Research and Development

Founded in 1981, TATA Research Design and Development Center, the R&D wing of TCS, focuses on applying science and technology to industry in particular and society in general. The most critical input to TCS' consulting practice, TRDDC, studies and invents leading-edge technology tools and engineering techniques, and improves intellectual property that is applied to commercial projects all over the world.

Challenges that lie ahead

Strengthening of the Indian Rupee is the major concern for part of the business that has foundations in North America. The current US recession is the next big IT challenge that TCS faces.

Maintaining such a large work force can never be a simple task. Also cutting down on the large attrition rate lies ahead as a major challenge.

Some Recent Achievements

- **IDC-DataQuest Best Employer** in IT Services in India for 2007.
- **Top position in 2007 "Global Services"** 100 'Top 10 Best Performing IT Services providers' category.
- **Tops the DataQuest DQTop20 list** of IT Services providers in India for 2007. **Banker Technology Award 2003** for its hallmark project CDSX executed for the Canadian Depository for Securities Limited (CDS) (June 06, 2003).
- **Highest number of CSQAs in any single organization in the world** (Quality Assurance International, April 2003)
- **Lotus Beacon 2003** Award from IBM for SmartGov Solution
- (January 27, 2003).
- Ranked #1 in **Dataquest IDC CSA** survey for CIO's choice of IT services companies in India(Dataquest, January 15, 2002)
- **Rajiv Gandhi National Quality Award** for TCS Sholinganallur Center
- (December 30, 2002)
- **Best IT Consultant** (Asia Pacific) for the Year 2002-03, Global Finance (January 03,2002)
- **Silicon India Company** of the Year 2002 (November 16, 2002)
- **Dataquest** Top Indian Software Company for 2002
- **Ranked 12th** largest IT Consulting Company (Consultants News, July 2002)

1.1.2 Quality Policy

TCS follows a continuous quality enhancement process, and as of today, fifteen centers – Qwest, HP, Sholinganallur, Ambattur and GEDC centers in Chennai, SEEPZ in Mumbai, Gurgoan II, Noida, and Gulab Bhavan in Delhi, and Bangalore, Calcutta, Lucknow, Pune, Hyderabad and Ahmedabad operations have all been assessed at Level 5 maturity of the Software Engineering Institute's Capability Maturity Model (SEI-CMM) scale. All major development centers belonging to TCS are ISO 9001 certified. TCS's Quality Management System also has ISO 9001 compliance.

- Organization wide ISO 9001 certification
- 15 Centers at SEI CMM Level 5 - Qwest, HP, Sholinganallur, Ambattur, GEDC(Chennai), Gurgaon II, Noida, Gulab Bhavan(Delhi), SEEPZ, Calcutta, Bangalore, Lucknow, Pune, Hyderabad and Ahmedabad
- For SEI CMM – TCS has 40 Assessors, 3 Lead Assessors and 1 People CMM Lead Assessor.
- First organization in the world to be assessed at PCMM Level 4 – Noida, Gurgaon II(Delhi), Kolkata and Tidal Park(Chennai).
- Adopting Six Sigma Practices (GE Center Bangalore) – TCS has 12 Six Sigma Black Belts.

The management of a Quality Management System in form of IQMS or Integrated Quality Management System is widely helpful in dealing with the following issues:

- Change Requests from Clients.
- Maintenance of Quality, Process and Standard Manuals.
- Work Assignments.
- Time Sheet submissions by employees.

Another aspect of the TCS quality policy is the presence of three phases of Quality Assurance for each deliverable be it a class diagram or a small snippet of code:

(I) Internal QA: Involves review of the deliverable by a person from the same team or module.

(II) External QA: Involves review of the deliverable by a person not from the project team.

(III) Final QA or Terminal Audit.

1.1.3 Communication

The project was undertaken at Tata Consultancy Services, Gandhinagar and TCSL can be contacted at the following:-

TATA CONSULTANCY SERVICES LTD
5TH & 6TH FLOOR, INFOTOWER-1,
INFOCITY, CH-0,
GANDHINAGAR 382 009
Phone: 91-79-6671 2600
Fax: 91-79-6671 2601

1.2 THE SYSTEM

1.2.1 Definition of the System

The system named "SYSTEM FOR ADMINISTRATION OF CLIENT REQUESTS" is being developed for a financial organization. As the name suggests the system develops an interface between the firm and its customers. Using this system, the customers and clients can place appropriate requests, define tasks and set request status. The system would also allow various facilities to the organization like appropriate assignment of requests to personnel and manipulating the status of these assigned requests and tasks.

1.2.2 Project Development Model

A software development model is a simplified description/abstract representation of a software process which is presented from a particular perspective. A development model for software engineering is chosen based on the nature of

the project and application, the methods and the tools to be used, and the controls and the deliverables that are required.

All software development can be characterized as a problem solving loop in which four distinct stages are encountered:

- Status Quo: Represents the current state of affairs.
- Problem Definition: Identifies the specific problem to be solved.
- Technical Development: Solves the problem through application of some technology.
- Solution Integration: Delivers the results like documents, programs, data, etc. to those who requested the solution.

There are many software development models. SACR follows the spiral model.

SPIRAL MODEL

The spiral model of software development and evolution represents a risk driven approach to software process analysis and structuring. This approach incorporates elements of specification-driven, prototype-driven process methods, together with the classic software life cycle. It does so by representing iterative development cycles as an expanding spiral, with inner cycles denoting early system analysis and prototyping, and outer cycles denoting the classic software life cycle. Every iterative cycle of the spiral model consists of the following:

- Customer Communication
- Planning
- Risk Analysis
- Engineering
- Construction and Release
- Customer Evaluation

The evolutionary process starts from the core of the spiral from the customer communication region and revolves in a clockwise direction as if it was coming out of the spiral.

The spiral model is considered as one of the most popular system process flow model as it contains the repeated execution of SDLC till the completion of the process. Some of its benefits are:

- Focuses attention on reuse options.
- Focuses attention on early error elimination.
- Puts quality objectives upfront.
- Integrates development and maintenance.
- Provides a framework for hardware/software development.

1.2.3 Concerned Audiences And Users

Users of the system SACR are as follows:-

Assistant: An assistant is a person who registers a proceeding into the system.

Consultant: A consultant is a key person in this system. A consultant is responsible for assigning a status to any proceeding like accepting it, rejecting it, retaining it i.e. .keeping it on hold etc.

Supervisor: A supervisor can just take an overview of the status of various proceedings. Unlike a consultant, the supervisor himself cannot mark the status of a proceeding.

1.2.4 Purpose and Objectives

The SACR (System for Administration of Client Requests) has a primary purpose of providing all round support to the customer. It fulfills the following objectives:

- Functionality harnessed by the customers
    - Searching a Type of Proceeding
    - Adding a new Type of Proceeding
    - Modifying a Type of Proceeding
    - Disabling a Type of Proceeding

- Functionality harnessed by the organization
    - Searching a Type of Consultant
    - Adding a new Type of Consultant
    - Modifying a new Type of Consultant
    - Disabling a Type of Consultant
    - Adding personnel to a Consultant
    - Searching a Type of Task
    - Adding a new Type of Task
    - Modifying a new Type of Task
    - Disabling a Type of Task

1.2.5 Proposed System

SACR is an integral part of a project on which the entire team of project trainees would be working.

The following example gives a peek into the kind of layers at which a proceeding i.e. a request will be progressed and handled.

SCENARIO: Customer wants to avail a loan.

PROCEEDING: Avail Loan
This layer deals with the customer actually wanting to lodge this request into the system. Now it's the system which will inquire about all the possible details and pass the proceeding to the later phases. This layer actually is the most abstract and gives the very basic detail about the request.

PROCEEDING IN BRANCH: Enter specific loan type

This phase involves the customer specifying what kind (BRANCH) of loan he/she is applying for. This may be of the kind EDUACTIONAL, MEDICAL, HOUSING, ARTICLE. This layer gives a direction to handling of the request.

PROCEEDING IN OPERATION: Enter details of the loan.

This phase involves the customer specifying the specific details (OPERATIONS) of the loan like duration, applicant status, salary status, etc. This is the stage when the minutest of details of the request are considered.

TASK: Breaking the entire process of loan assignment into smaller tasks.

This phase starts when all details have been submitted by the customer. It involves breaking down the entire process of loan dissemination into manageable tasks. The following can be a possible set of tasks:

- Document Gathering
- Document Verification
- Amount and EMI calculation
- Legalities
- Amount Dissemination
- Acknowledgements

CONSULTANT: Assignment of Tasks to specific departments.

This phase involves assigning the task to specific departments of the organization. Such as the "Amount and EMI calculation" task to the Finance Department and the "Amount Dissemination" task to the Operation Department. Specific personnel of the consultant will handle the specified task.

So administration of a proceeding would involve manipulation of tables which contain information of the above phases.

The following requirement statements and the business processes analysis describe the proposed system:

Functional Requirements

- Search for Types of Proceedings
  The system may have certain "Types of Proceedings" defined initially. So there must be a facility through which specific "Types of Proceedings" can be searched. The search criteria may be based on the basis of the name of that particular "Type of Proceeding".

- Add Types of Proceedings
  The system must allow for creation of a new "Type of Proceeding". The required data should be the name of the "Type of proceeding", description of the "Type of Proceeding" and the date from which the "Type of Proceeding" has come into existence. Also the system should generate a unique identifier for each new "Type of Proceeding" that is being created.

- Modify Types of Proceedings
  The system must allow for modification of an already existing "Type of Proceeding". The changes can be made to all the associated data namely the name of the "Type of Proceeding", description of the "Type of Proceeding" and the date from which the "Type of Proceeding" has come into existence. This function would involve the following:

  - Set the original proceeding to inactive.
  - Insert a new proceeding with the modified details.
  - Set the original proceeding in branch to inactive.
  - Insert a new proceeding in branch that points to the new proceeding inserted.
  - Set the original proceeding in operation to inactive.
  - Insert a new proceeding in operation that points to the new proceeding in branch inserted.
  - Set all the tasks associated to the original proceeding in operation to inactive.
  - Insert a new set of tasks and make them point to the new proceeding in operation.

- Set all the relationships of the particular task and consultants to inactive.
- Insert a new relationship between the task and the consultants involved.

- Disable Types of Proceedings

  The system must allow for disabling of an already existing "Type of Proceeding". It would actually involve setting a field namely "INACTIVE" to "true" stating that the record is no longer present. This function would involve the following:

  - Set the proceeding to inactive.
  - Set the proceeding in branch to inactive.
  - Set the proceeding in operation to inactive.
  - Set all the tasks associated to the original proceeding in operation to inactive.
  - Set all the relationships of the particular task and consultants to inactive.

Non - Functional Requirements

- Reliability

  Reliability of the system is of primary importance. As the system is internet based and would be accessed many times by various different clients for various different purposes, it should entirely robust and reliable.

- Maintainability

  The system should be designed to be easily maintainable and get the least complaints from the users and would guarantee high customer satisfaction and minimum downtime.

- Adaptability

  The system must be entirely adaptable and should easily gel with the parent modules without causing much of rework or displacement.

- Extensibility

  The system should be designed to be extensible to changes. Changes might be a result of

  - User requirement change.
  - Compliance to follow some new company policy.

Performance Requirements

The system is to be used as a web application and targets a wide community of users which are the customers all over. The system is expected to be robust i.e. up and running all the time. The speed factor of the system is as critical as the reliability and availability factor. As the system is to work as a networked application in a client server environment, the system should have enough speed to deliver the required information to fill in for the lags due to network problems.

Business Process Analysis

The system is to be developed taking care of certain business rules that form in a way a major development directive and system descriptor. Some of the important business rules are as follows:

- **Any deletion in the database should be soft in nature.**

  The rule specifies that a record shouldn't be physically deleted from the database and hence should be deactivated. This would be of help during company audits and other quality checks.

- **Session Removal should be done appropriately while GUI navigation.**

  This rule refers to the practice of session removal and session retain while navigating through the various screens depending on the requirements.

- **Facility provided by the technology employed should be utilized to its maximum.**
  This refers to strict employment of the tools and technology being used.

- **Development should be in accordance to the Software Design Document.**
  This rule stresses the importance of the Software Design documents. They are the main source of requirements for off site developers. And depending on various versions of the SDD the change requests are recorded. Finally the extra effort involved in solving these change requests is recovered from the client.

- **All deliverables should undergo a self review by the developer.**
  This business rule stresses on the rechecking process to be carried out by the developer. This implies that once the deliverable undergoes QA it should be with minimum errors and in turn involve minimum rework.

1.3 PROJECT PROFILE

1.3.1 Project Title

Title of the Project is:

SYSTEM FOR ADMINISTRATION OF CLIENT REQUESTS

1.3.2 Scope of the Project

This project named "System for Administration of Client Requests" is designed as a sub-module for a larger module. The focus is on the parameters concerning the employment of this sub-module and its scope of usage in the parent module. SACR develops an interface between the firm and its customers. Using this system, the customers and clients can place appropriate requests, define tasks, set request status and allow for appropriate assignment of requests to personnel and manipulating the status of these assigned requests and tasks. It also has a wide scope if it is to be employed in other similar environments.

1.3.3 Project Team

- TCS proposes the following team structure for successful implementation of the project.
- The TCS team will be headed by a Project Manager and will be supported by Module Leaders and team members.
- The team has a multi-disciplinary skill set comprising of domain experts, system administrators, system analysts, Networking Specialists and Internet technocrats.

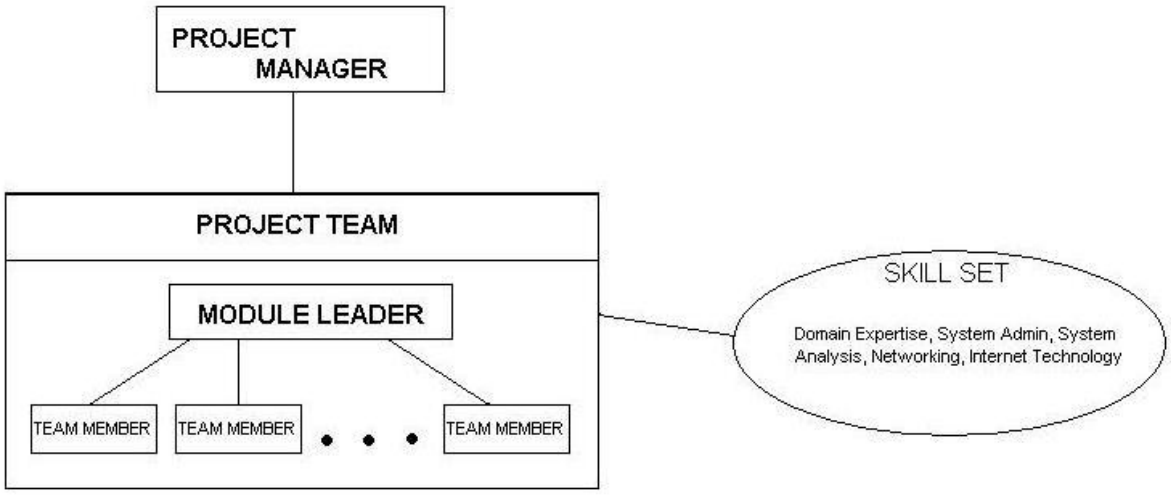


Figure 1.1 Project Team Structure

Project team specifically for SACR:

- SACR comprised of 3 Project Trainees each handling the one of the three aspects:
  - Task
  - Consultant
  - Proceeding
- SACR was a sub module of the larger module Control of Proceedings which in turn comprised of several of the employees working on the project.

1.3.4 Hardware/Software Environment in the Company

Table 1.1 Hardware/Software Being Used

Hardware	RAM	1.5 to 2 GB
	Hard Disk Capacity	80 GB – 160 GB
	Processor	3 GHz
Software	IBM WebSphere Integration Developer (Version 6.0.1)	
	PL/SQL Developer (Version 5.1.3)	
	Crystal Report Designer XI	
Database	Oracle 10g	
	HSQLDB	
Version Management	Microsoft Visual SourceSafe (Version 6.0c)	
Documentation and Change Management	Microsoft Office 2003, XP	

2. SYSTEM ANALYSIS

2.1 FEASIBILITY STUDY

An important outcome of the preliminary investigation phase of the System Development Life cycle is the determination that the requested system is feasible. The following facets describe the feasibility:

- Operational Feasibility
- Technical Feasibility
- Financial and Economic Feasibility
- Knowledge about handling infeasible projects

2.1.1 Operational Feasibility

The proposed system is beneficial if and only if it can be turned into an information system that will meet the operating requirements of the organization. This facet of feasibility asks: if the system will work when it is developed and installed and are there any major barriers for system implementation and many more issues.

Issue 1: Is the user happy with the new system….
The organization for which the system is being constructed wants its customers to lodge in the requests pertaining to their problems into this system. So the organization is satisfied with the SACR. Moreover the organization will also harness various functions from the system and hence the user is happy with the new system being introduced.

Issue 2: Have the end users been involved in the development and requirement elicitation phase…
There has been an active participation of the end-users in the requirement gathering phase and hence chances of user inacceptance are low. Because the success of the system depends on the review of the end users and if there has

been an active participation from their side it leads the system to a better place. This part is explained in detail in the Requirement Elicitation part of the report.

Issue 3: Will the user be affected in any undesirable manner after the system is installed….
This lies at the far end of the list of possibilities. The SACR actually facilitates its own use. The user will be encouraged to use the system to his ease. Even on the part of the organization the assignment of requests breaking into tasks will all take place with the absolute ease.

This facet of feasibility i.e. Operational Feasibility, hence concentrates on many such issues and gives their solutions.

2.1.2 Technical Feasibility

The investigation is done mainly keeping in mind this aspect of feasibility. The analyst has to keep in mind while performing requirement elicitation that the requested work flow is what is feasible technically. The following issues can be raised while the investigation is underway:-

Issue 1:
Can the work for the project be done with the current equipment, existing software technology and available resources….
In the case of this project if the analyst (person from TCS) feels the answer to this question is on the negative side, he can ask the client to deliver the required kind of technology, hardware and resources or either deal for the same in monetary terms. So this factor can never be a hinderance in the project being feasible.

Issue 2:
Does the proposed equipment have the capacity to process and hold the kind of data involved…..
This issue is mainly based on the developers' vision towards the tool and propensity of the tool also. In the development of the SACR, a set of handsome hardware and software requirements was stated surely keeping in mind that the

proposed technology will not only be at the developer's ease but also have the required propensity to hold, process, disseminate and terminate data.

2.1.3 Financial and Economic Feasibility

This facet of feasibility is of paramount importance in development of any software for any particular company. TCS is having licensed versions of the required software (IBM WebSphere Integration Developer, PL/SQL Developer, Microsoft Visual SourceSafe, Oracle 10g). Hardware Costs mainly involve the cost of the extended RAM capacities and server maintenance. Maintenance costs mainly involve the cost of application and database server maintenance.

The issues that are raised here are:

Issue 1:
Are there sufficient cost benefits in creating the system?
The answer to this issue is yes for the reason that this system will not only enhance their customer interaction but result in widening of the customer base. Current customers can be benefited in many different ways with the system.

Issue 2:
Are the costs of system installation high and unmanageable?
This is again a rare possibility as the technologies involved i.e. J2EE and MVC architecture are actually platform independent in nature. Hence, whatever be the available platform these technologies will gel easily and in turn make the system installation easy.

2.2 REQUIREMENT ANALYSIS

2.2.1 Fact- Finding Techniques

The client in most cases is not sure of what exactly is desired and has a poor understanding of the computing environment. But the client involved in this case has an entire Information Technology department of its own. But it has believed

to the fact of outsourcing the entire development to an outside organization. Being so very rich with knowledge about the technology involved, the requirement dissemination process is very streamlined. But again the basic fact finding techniques do come into play and the ones employed in the project are as follows:

- Inception of the Project
- Basic Elicitation
  - Problems of Scope
  - Problems of Understanding
  - Problems of Volatility
- Elaboration
- Negotiation
- Specification
- Validation
- Management (Continuous)

The following techniques are present unambiguously throughout the project and possess enormous power with regard to requirement gathering.

2.2.1.1 Interviews

The requirement analysis phase begins after the inception of the project.
The first phase of interviews is mainly a kind of informal discussions with the client. In this phase the analysts who are the evangelists in the process of requirement elicitation generally do the following:

- Ask a set of Informal Context Free Questions regarding the system.
- Talk through with the client to know his intention with regard to the project.
- Define a business case for the idea along with the performance of certain kind of market analysis.
- Identify a working description of the project's scope.

The later phases of interviews involve the following kind of facets:

- Discussion on the Division of the entire thing into manageable and doable modules.
- Module wise interviews with the various personnel involved.
- Certain kind of debatable presentations which may be clubbed with brainstorming or prototyping sessions.

This mode of requirement gathering is the one that provides the maximum amount of information regarding the project and hence is used very effectively. This mode can turn into all various forms ranging from strict one room interviews to large debatable discussions.

2.2.1.2 Questionnaire

This mode of requirement elicitation is generally employed during change management and while laying out basic system explanations.
Questionnaires used in the project are framed keeping into mind the following things:

- Amount and the kind of information to be extracted through this channel.
- The kind of stake holder to whom the questionnaire is addressed.
- The reusability and abstractness of these questionnaires.

2.2.1.3 Record Review

The records analyzed by me were mainly the following:

- Software Design Document
  This gave me the actual requirements of the GUI plus the backend logic right till statement of logical queries which may be employed in some or the other form. It also incorporated the sample GUIs so that any changes to the prototypes submitted earlier can be checked and tracked.

- Data Dictionary of entire module
  I analyzed the entire Data Dictionary to find out the correlations that existed between the necessary tables and other unwanted ones. And hence I could out sort out the ER statement from the Data Dictionary provided.

- Technical SRS (with Business Analysis)
  This was a typical SRS that gave me the specific requirements along with the Business rules that need to be employed.

- Class Diagrams
  The class diagram made me understand the entire architecture that was employed and allowed me to extend it in my system.

- Collaboration Diagrams
  The collaboration diagram helped me understand the system flow and similarly I could employ the same techniques in my diagrams.

2.2.1.4 Observation

This is also the method employed very widely in the project being developed. The developers working onsite generally engage in the observation of the following things:

- Work Environment of the organization.
- The technical expertise of the employees of the organization.
- The volume of customers entertained.
- The kind of system expected.
- The resistance in the organization due while the organization gets the system installed.
- The usage of any of the available systems.

During the continuous management phase that starts once the system is installed and is running the observation regarding system usage, system inconveniences and system benefits is carried out.

2.3 CONTEXT LEVEL DIAGRAM

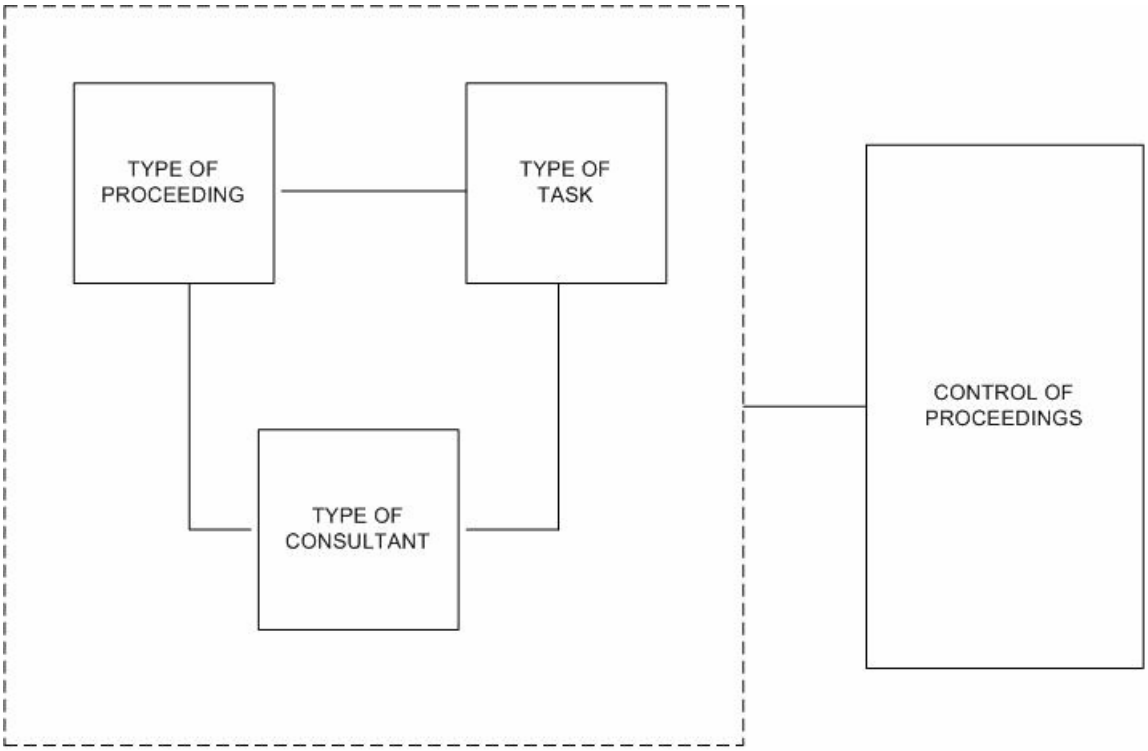


Figure 2.1 Context Level Diagram

Context Level Diagram

Name: Control of Proceedings

Involved Entities:

Module → Control of Proceedings

Sub Modules → Type of Consultant, Type of Task and Type of Proceeding.

Associations Involved:

Sub modules "parts of" Module

Sub modules "interconnected"

2.4 DATA FLOW DIAGRAMS

2.4.1 Level I Diagrams

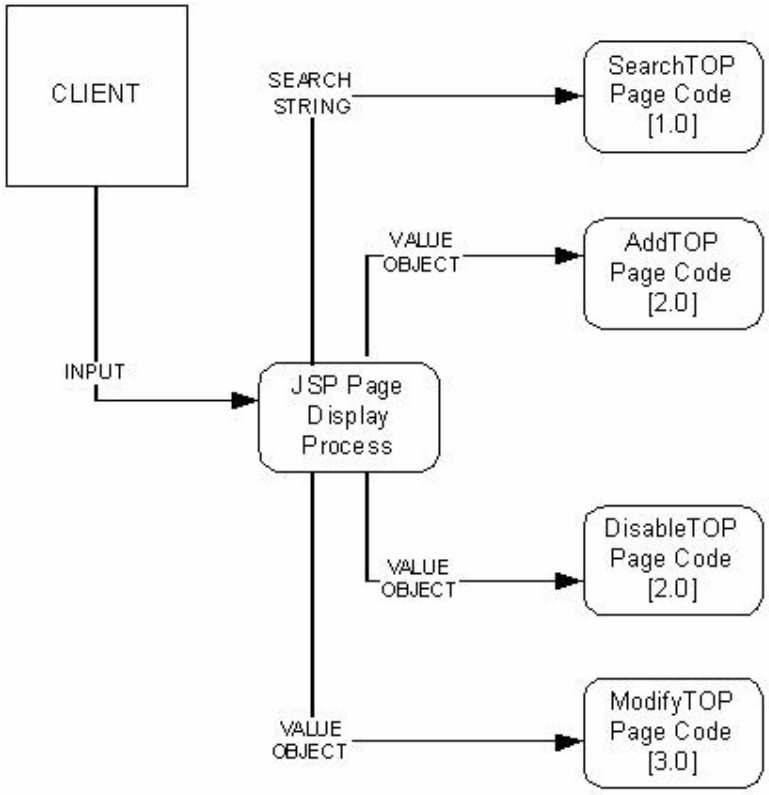


Figure 2.2 Level I DFD

2.4.2 Level II Diagrams

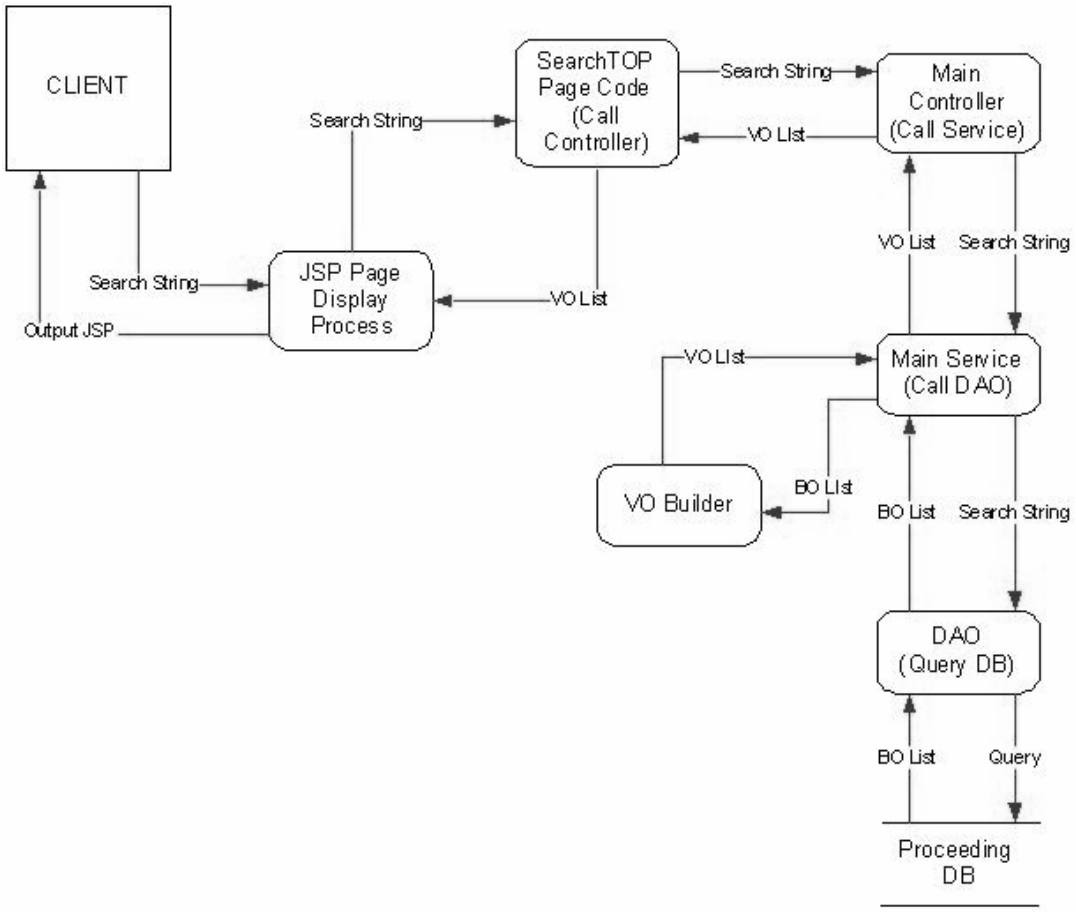


Figure 2.3 Level II DFD – Search Proceeding

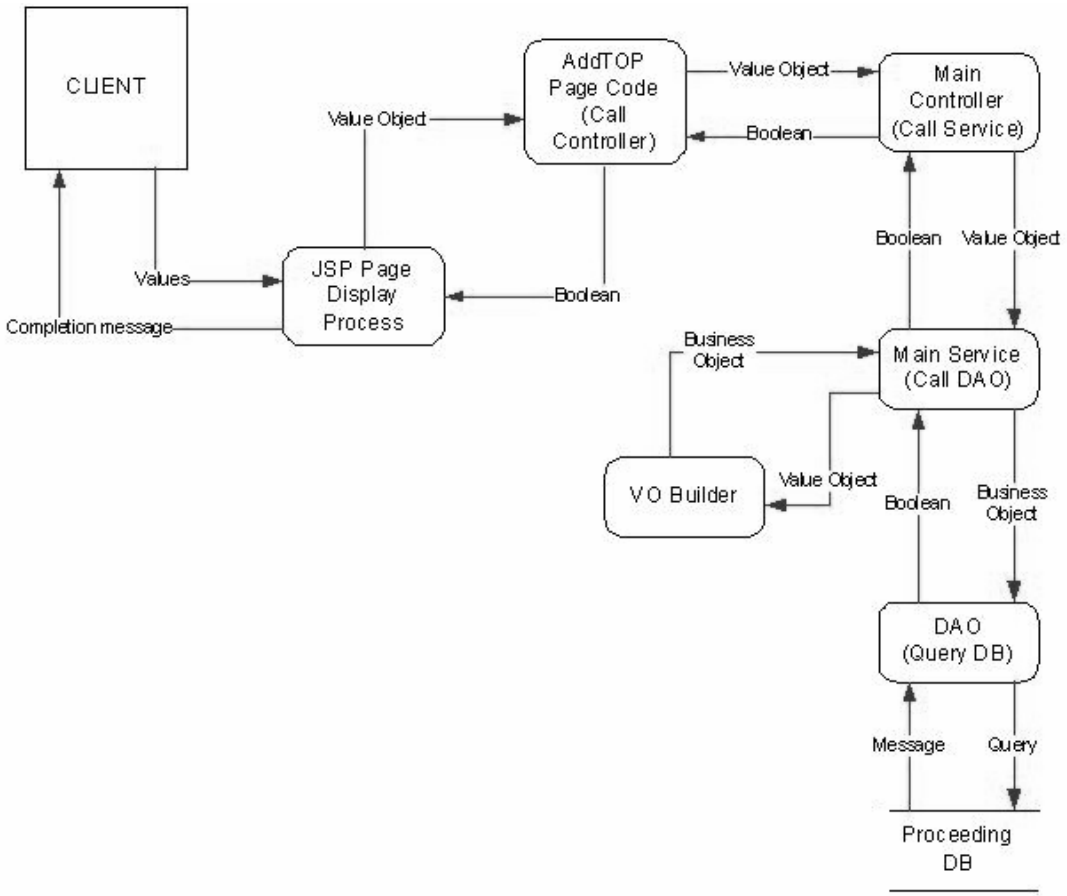


Figure 2.4 Level II DFD – Add Proceeding

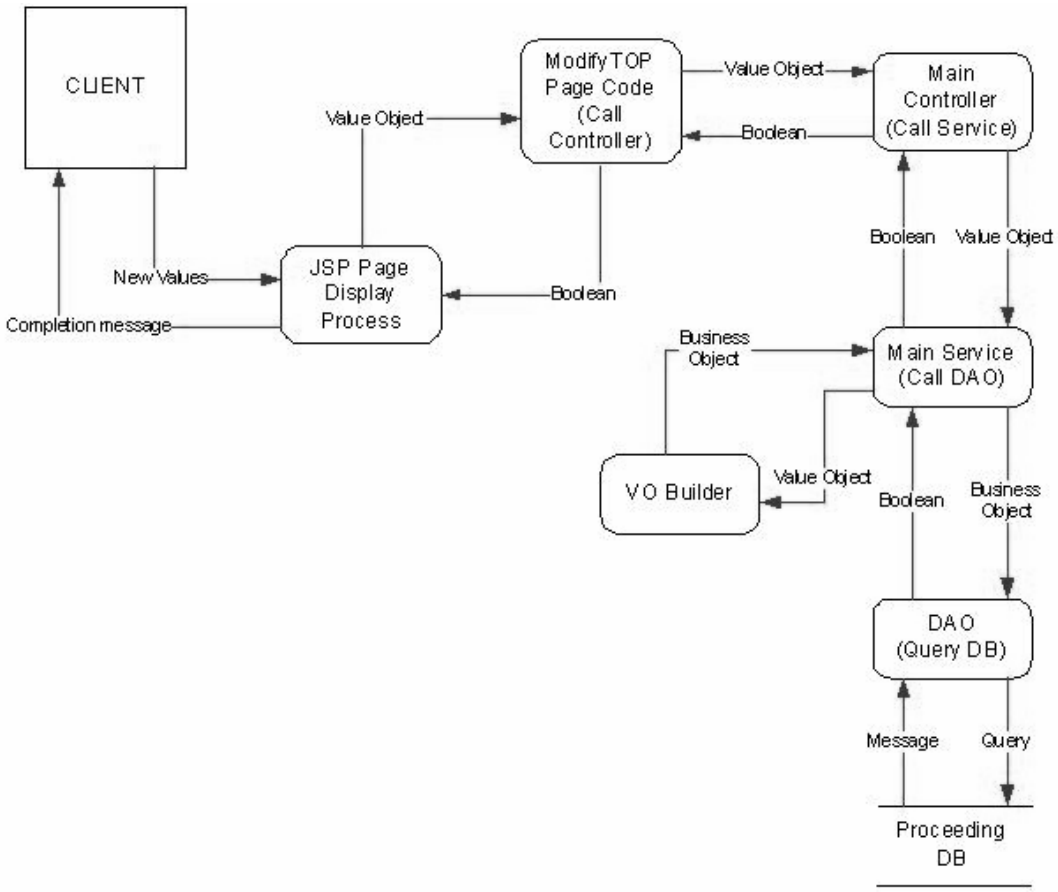


Figure 2.5 Level II DFD – Modify Proceeding

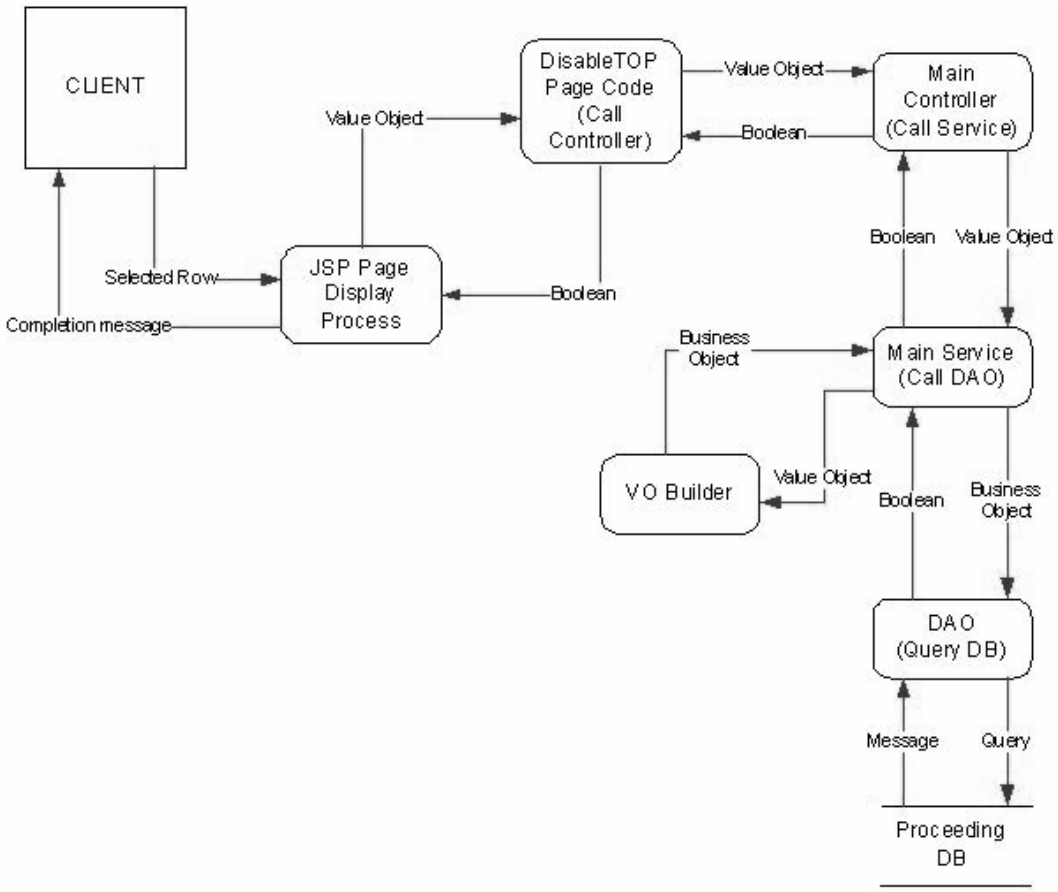


Figure 2.6 Level II DFD – Disable Proceeding

3. SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

3.1.1 Model View Controller Architecture

The Model-View-Controller architecture is a widely-used architectural approach for an application where there is user interaction. The basic idea of Model-View-Controller (MVC) architecture is the separation of the user interface of an application from its domain logic. It divides the functionality among objects involved in maintaining and presenting data to minimize the degree of coupling (relationship) between the objects. Basically, it separates business, navigation and presentation logic.

MVC is organized into three separate areas:

- Model – Model includes the data of the application, along with the business logic that defines how to change and access that data. The Model can be shared among any number of view and controller objects. The Model object knows about the data that need to be displayed. It also knows about all the operations that can be applied to transform that data. However, it knows nothing about the GUI and the manner in which the data is to be displayed.
- View – View represents the means of presenting the Model's data to the outside world. This could take the form of a GUI, generated speech, audible tones, printouts, or even non-user oriented output, such as turning on an air conditioner. A view renders the contents of a Model. It accesses the enterprise data through the model and specifies how that data should be presented.
- Controller – Controller represents a means of gathering user input and providing feedback to the model, handle requests, and control application flow. Controller contains no business logic and deals strictly with exposing the methods contained in the Model to the user through the View. A Controller translates the user interactions with the View into actions to be performed by the Model. Controller is represented by a Servlet. Controller centralizes the logic for dispatching requests to

the next view based on the request URL, input parameters, and application state.

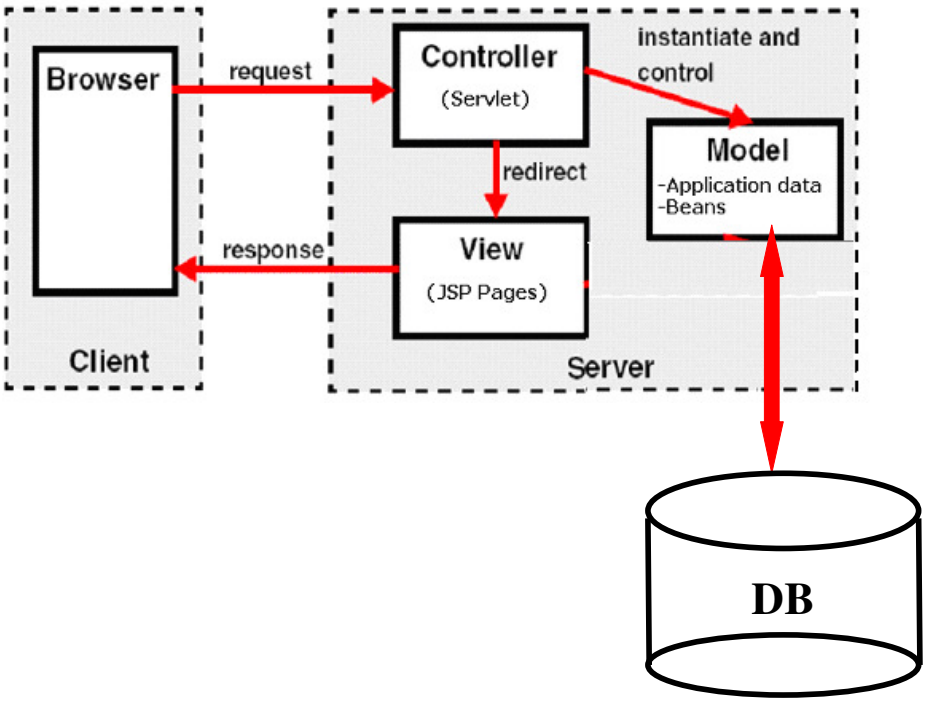


Figure 3.1 MVC Architecture

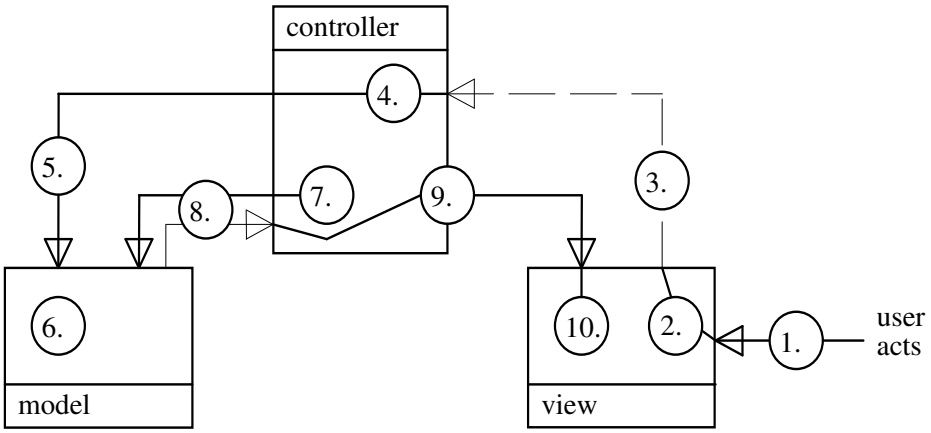


Figure 3.2 Common MVC Architectural Flow

Steps of MVC Architectural Flow:

1. The end user manipulates the view.
2. The user's actions are interpreted by the view.
3. The view passes the user's actions to the controller.
4. Based on the action, the controller decides what should be done.
5. The controller delegates/calls the appropriate model to act upon the command.
6. The model acts independently by processing the inputs. Model usually accesses the database and performs some business logic on the data.
7. After the model is finished. The controller decides what view should be returned to the end user. Example, login controller decides the next view to display after a sign on very likely depends on: the user id and password contained in the request, the success or failure of the sign on operation, the URL the user was trying to access etc.
8. The controller gets all the information needed from the model.
9. The controller passes the information to the view.
10. The view displays the information in its own way.

Advantages and disadvantages

Overall, MVC design approach provides several key advantages over other approaches in interactive applications:

- Allows the decoupling of presentation logic and business logic.
- Separation of concerns makes the code cleaner and easier to understand.
- Eases maintenance burden. For Example, changes made to business logic are less likely to break the presentation (view).
- Facilitates multi-team development. Developers can focus on business code and UI designers can focus in building the view without worrying about the backend code.

The disadvantages of MVC architecture:

- Increases design complexity. MVC introduces some extra classes/code due to the separation of Model, View, and Controller.
- Requires large amount of configurations and architecture work.

3.1.2 Project Architecture (Detailed)

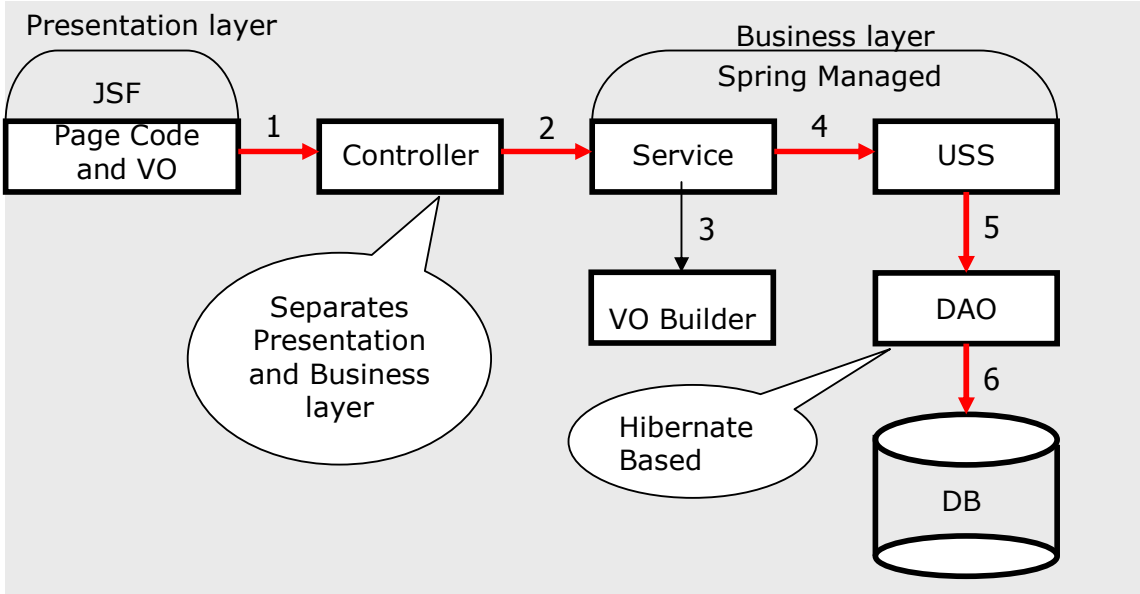


Figure 3.3 Project Architecture

The project involves three layers:-

Presentation Layer:

JSF, Page Code and VO.

This layer comprises of a set of Page code, VO and jsp pages. Every jsp page has an associated VO which mainly contains the properties that are bound to the components on the jsp page. Page code acts as a container for a set of methods that are to be executed as a result of actions that take place on the screen. Page code is responsible for making a call to a method in Controller.

Controller:

Controller is responsible for separating the presentation layer and the business layer. Controller ensures that presentation layer is unaware of the business logic. Controller does not contain any business logic. It simply invokes a method in Service.

Business Layer:

Service:

Service is entirely spring managed which means that an object of service would be returned to controller by "Inversion of Control". Service is responsible for making a call to Use-case Specific Service and also converts VO to BO (Value Object to Business Object). Basically, a business object is an object which represents as enterprise data. It can be taken to be an object of a persistent class, i.e. a class which gets stored in a database. Since, only a business object can be saved into a database, conversion from VO to BO is inevitable.

VO Builder:

VO Builder is responsible for converting VO to BO, when VO is passed from the front end. Also, it converts BO to VO when either a single BO or a list of BOs is passed from DAO.

Use-Case Specific Service:

A Service is split in a set of use-case specific services. This facilitates parallel development of the components of a module. Since, SACR has three components – Consultants, Tasks, Proceedings, would have three different use-case specific services.

Persistence Layer:

Data Access Objects:

Data access objects represent the persistent layer. Data access objects are responsible for dealing with database.

Advantages:

Enhances Maintainability:

Since, the presentation logic, business logic and the persistence logic are clearly separated, it becomes easier to debug and maintain the code.

Facilitates multi-team development:

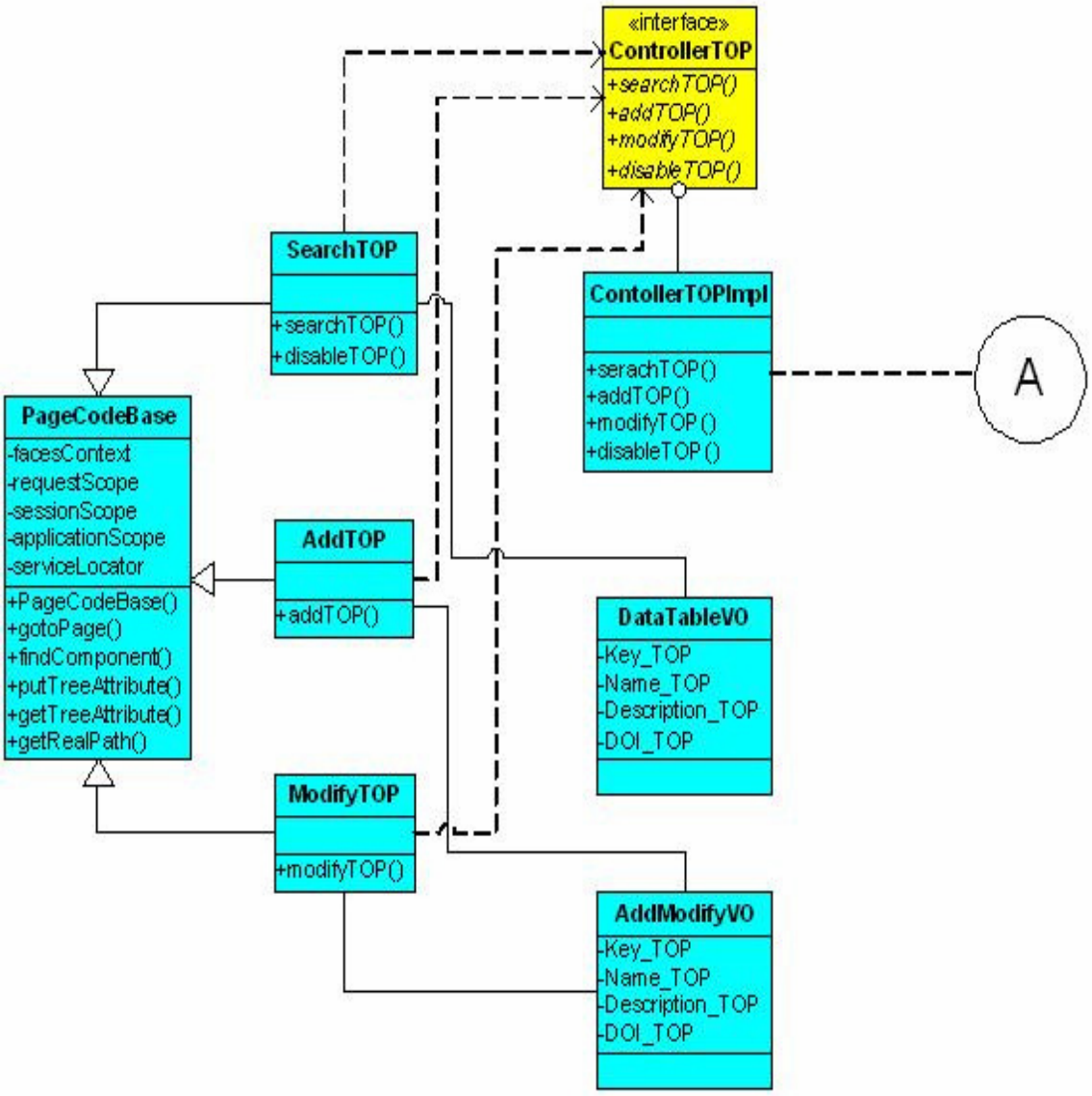
Developers can focus on business code and UI designers can focus in building the view without worrying about the backend code

Disadvantage(s):

Increases design complexity:

Such a layered architecture introduces large number of classes. Hence, there are a large number of files to be dealt with even in a single module.

3.2 CLASS DIAGRAM



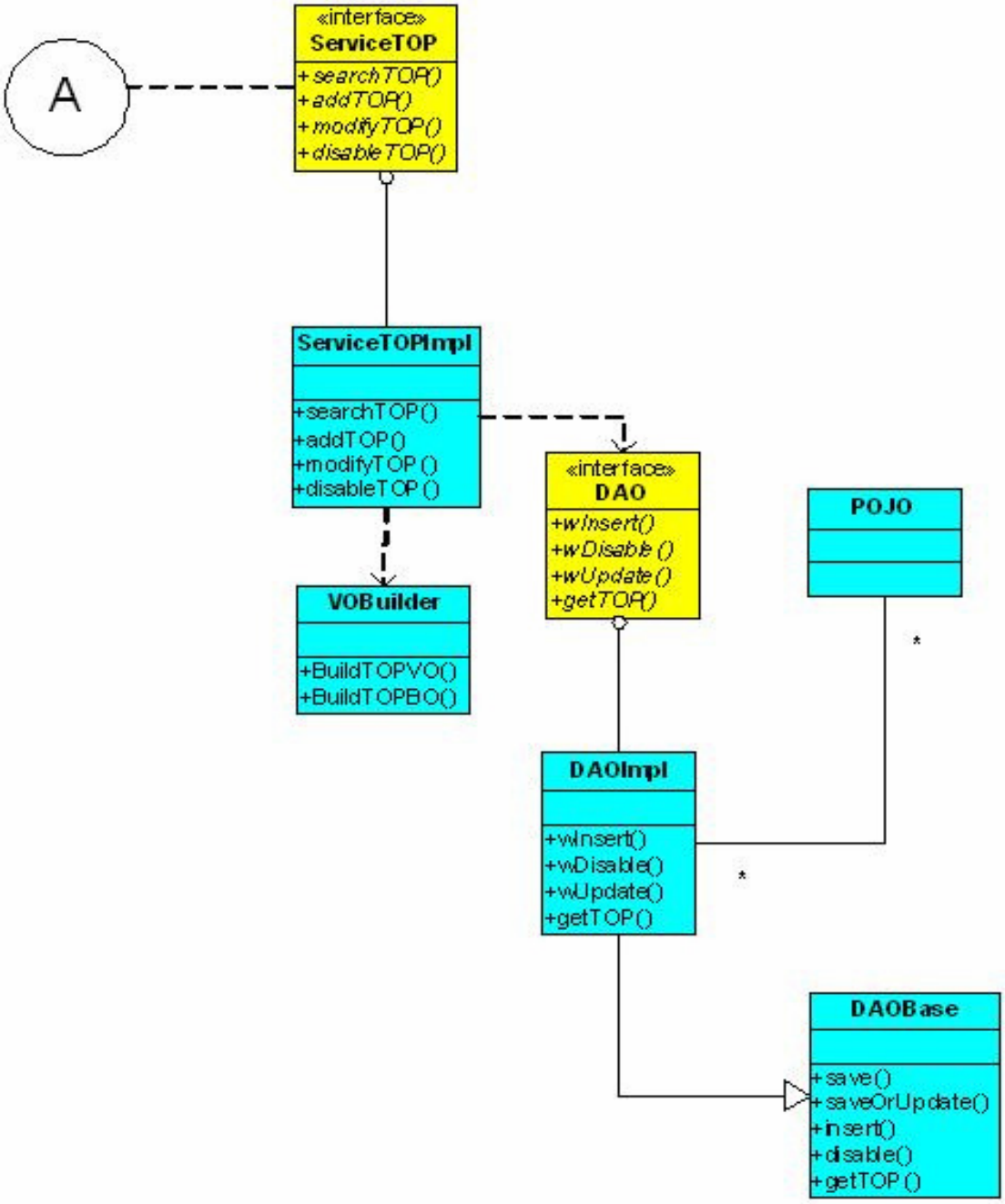


Figure 3.4 Class Diagram

3.3 SEQUENCE DIAGRAMS

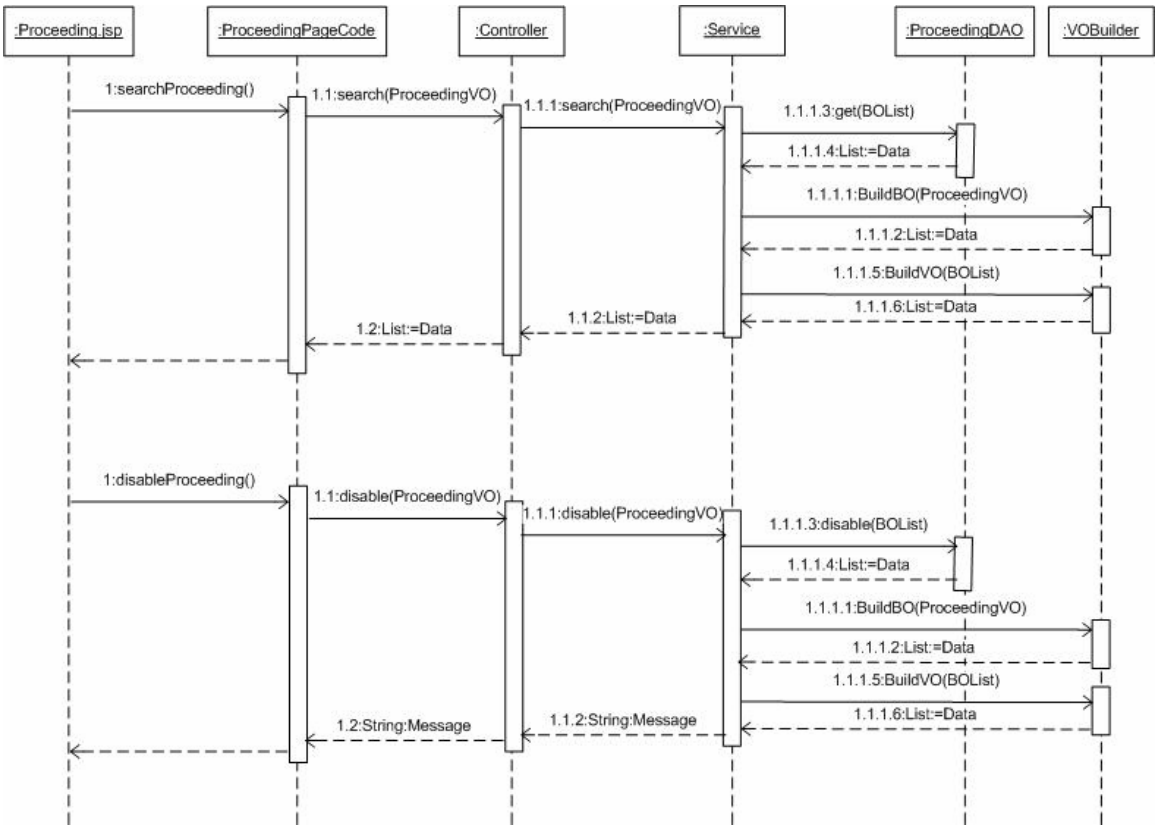


Figure 3.5 Sequence Diagram – Search and Disable Proceeding

Sequence Name: Search Type of Proceeding and Disable Type of Proceeding.
Sequence Description: Shows the sequential interaction among entities while searching and disabling a type of Proceeding.

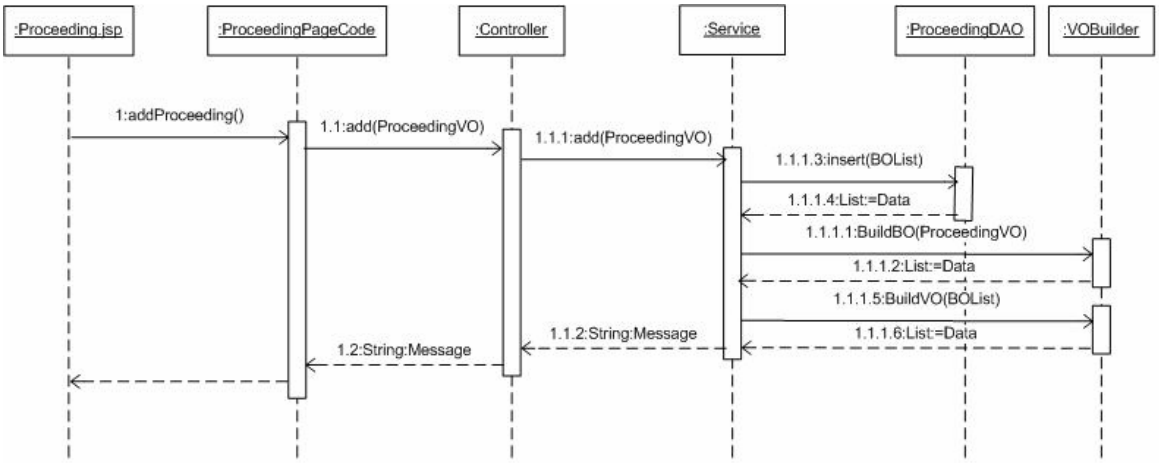


Figure 3.6 Sequence Diagram – Add Proceeding

Sequence Name: Add Type of Proceeding.

Sequence Description: Shows the sequential interaction among entities while adding a type of Proceeding.

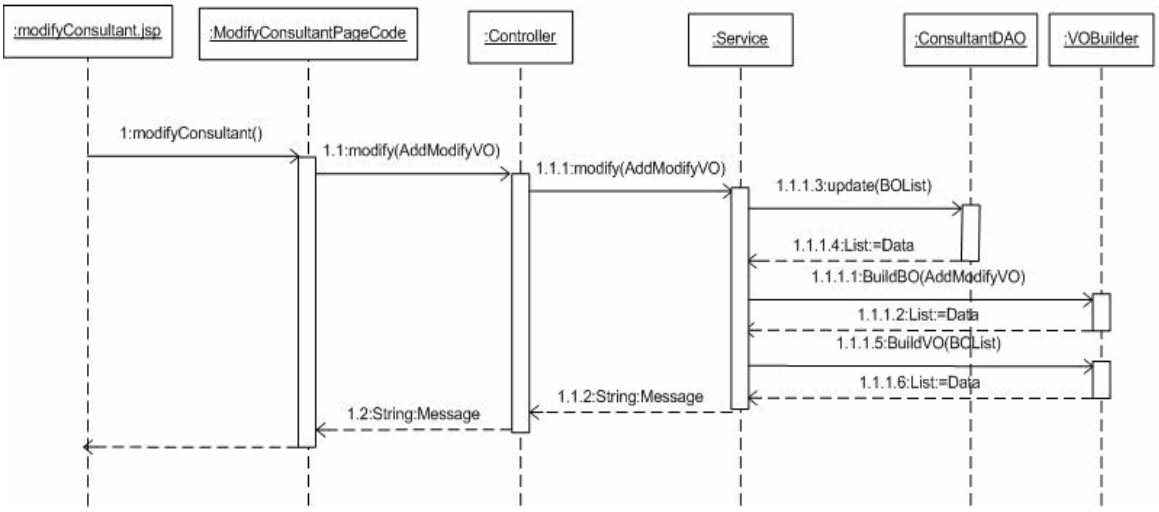


Figure 3.7 Sequence Diagram – Modify Proceeding

Sequence Name: Modify Type of Proceeding.

Sequence Description: Shows the sequential interaction among entities while modifying a type of proceeding.

3.4 DATA DICIONARY AND TABLE DIAGRAM

The following is the detailed Data Dictionary:

Table 3.1 List of Columns of the table T_TYPE_OF_PROCEEDING

Code	Comment	Data Type	Mandatory	Primary	Foreign Key
ID_TYPE_OF_PROCEEDING	Unique identifier of type of proceeding.	INTEGER	T	T	
CODE_TYPE_OF_PROCEEDING	Code of Type of proceeding.	NUMBER(4)	T		
NAME_TYPE_OF_PROCEEDING	Name of Type of proceeding.	VARCHAR2(100)	T		
DESCRIPTION_TYPE_OF _PROCEEDING	Description of type of proceeding.	VARCHAR2(256)	T		
DATE_OF_INITIATION	Start validity date of type of proceeding.	DATE	T		
DATE_OF_TERMINATION	Finish validity date of type of proceeding.	DATE			
INACTIVE	It indicates whether the	INTEGER	T		

	record is active or logically inactive.				
CREATOR_NAME	Saves user who creates the record	VARCHAR2(30)	T		

Table 3.2 List of Columns of the table T_PROCEEDING_IN_BRANCH

Code	Comment	Data Type	Mandatory	Primary	Foreign Key
ID_PROCEEDING_IN_BRANCH	Unique identifier of proceeding in branch.	INTEGER	T	T	
ID_TYPE_OF_PROCEEDING	Unique identifier of type of proceeding	INTEGER			T
NAME _PROCEEDING_IN_BRANCH	Name of proceeding in branch.	VARCHAR2(100)			
DATE_OF_INITIATION	Start validity date of proceeding in branch.	DATE	T		
DATE_OF_TERMINATION	Finish validity date of proceeding	DATE			

	in branch				
INACTIVE	It indicates whether the record is active or logically inactive.	INACTIVE	T		
CREATOR_NAME	Saves user who creates the record	VARCHAR2(30)	T		

Table 3.3 List of columns of the table T_PROCEEDING_IN_OPERATION

Code	Comment	Data Type	Mandatory	Primary	Foreign Key
ID_PROCEEDING_IN_OPERATION	Unique identifier of proceeding in operations	INTEGER	T	T	
ID_PROCEEDING_IN_BRANCH	Unique identifier of proceeding in branch	INTEGER	T		T
NAME_PROCEEDING_IN_OPERATION	Name pf proceeding in operation	VARCHAR2(100)	T		
DATE_OF_INITIATION	Start validity date of	DATE	T		

	proceeding in operations				
DATE_OF_TERMINATION	Finish validity date of proceeding in operations	DATE			
INACTIVE	It indicates whether the record is active or logically inactive	INTEGER	T		
CREATOR_NAME	Saves user who creates the record	VARCHAR2(30)	T		

Table 3.4 List of columns of the table T_OPERATION_TASK

Code	Comment	Data Type	Mandatory	Primary	ForeignKey
ID_OPERATION_TASK	Unique identifier of the relation between the proceeding in operations and the task.	INTEGER	T	T	
ID_PROCEEDING_IN_OPERATION	Unique identifier of proceeding in operations.	INTEGER	T		T

ID_TYPE_OF_TASK	Unique task identifier	INTEGER	T		T
INACTIVE	It indicates whether the record is active or logically inactive.	NUMBER	T		
DATE_OF_INITIATION	Date component allowing the selection of the validity start date of the relation between the proceeding in operation and the tasks.	DATE	T		
DATE_OF_TERMINATION	Date component allowing the selection of the validity finish date of the relation between the proceeding in operation and the tasks.	DATE			
CREATOR_NAME	Saves user who creates the record.	VARCHAR(30)	T		

Table 3.5 List of columns of the table T_TYPE_OF_TASK

Code	Comment	Data Type	Mandatory	Primary	Foreign Key
ID_TYPE_OF_TASK	Unique identifier	INTEGER	T	T	

	assigned to a task.				
ID_TYPE_OF_CONSULTANT	Unique Id of the consultant.	INTEGER	T		T
NAME_TYPE_OF_TASK	Name of the task.	VARCHAR2(100)	T		
CODE_TYPE_OF_TASK	Code assigned to a consultant.	NUMBER	T		
DESCRIPTION_TYPE_OF_TASK	Description of the task.	VARCHAR2(100)	T		
STANDARD_OF_TASK	Indicates the number of days needed to perform a task.	VARCHAR2(100)	T		
INACTIVE	Indicates whether the record is active or not.	NUMBER	T		
DATE_OF_INITIATION	Specifies the date of creation of task.	DATE	T		
DATE_OF_TERMINATION	Specifies the end date of task.	DATE			
CREATOR_NAME	Specifies the name of the user requesting the task	VARCHAR2(20)	T		

Table 3.6 List of columns of the table T_TYPE_OF_CONSULTANT

Code	Comment	Data Type	Mandatory	Primary	Foreign Key
ID_TYPE_OF_CONSULTANT	Unique identifier assigned to a consultant.	INTEGER	T	T	
NAME_TYPE_OF_ CONSULTANT	Name of the consultant.	VARCHAR2(100)	T		
KEY_TYPE_OF_CONSULTANT	Code assigned to a consultant.	VARCHAR(4)	T		
DESCRIPTION_TYPE_OF_ CONSULTANT	Description of the consultant.	VARCHAR2(100)	T		
INACTIVE	Indicates whether the record is active or not.	NUMBER	T		
DATE_OF_INITIATION	Specifies the date of creation of task.	DATE	T		
DATE_OF_TERMINATION	Specifies the end date of task.	DATE			
CREATOR_NAME	Specifies the name of the user requesting the task	VARCHAR2(20)	T		

The following are the sequences used:

Table 3.7 Description of SEQ_CODE_TYPE_OF_PROCEEDING

Name	SEQ_KEY_TYPE_OF_PROCEEDING
Object Type	Sequence
Primary Key	-
DBMS	ORACLE Version 10g

Table 3.8 Description of SEQ_ID_TYPE_OF_PROCEEDING

Name	SEQ_ID_TYPE_OF_PROCEEDING
Object Type	Sequence
Primary Key	-
DBMS	ORACLE Version 10g

The following is the detailed Table Diagram:

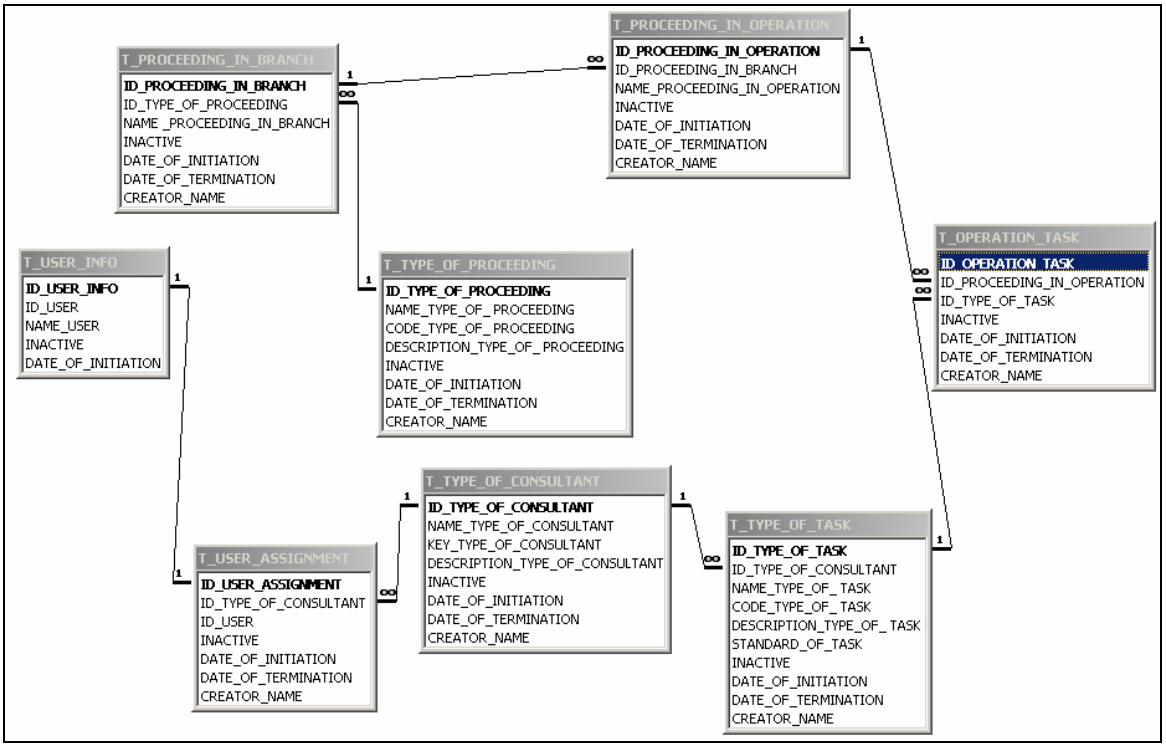


Figure 3.8 Table Diagram

The following are the screens with their description:

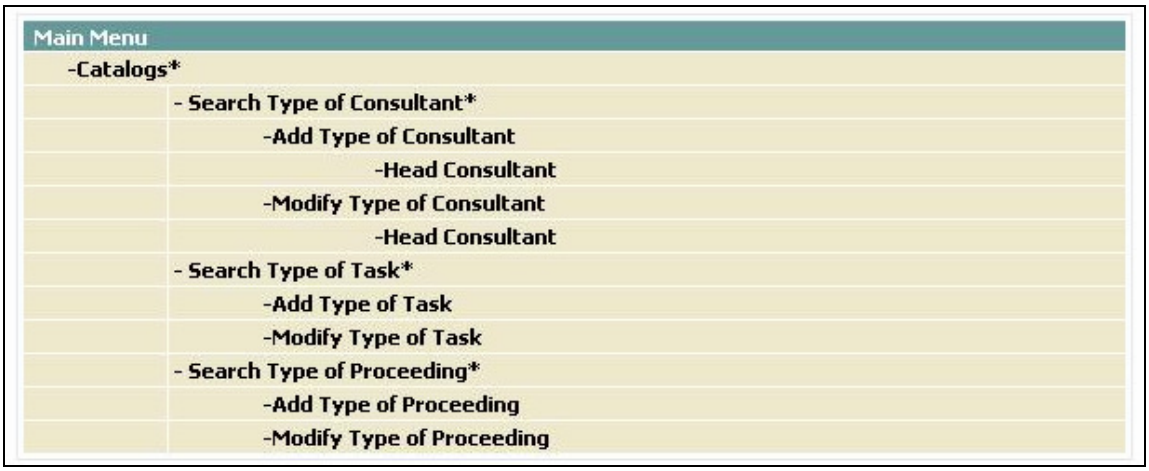


Figure 4.1 Screenshot of Main Menu

Screen Name: index.jsp

Description:

- This is the menu screen which is displayed to the user as soon he/she starts the application.
- This screen contains links to several other pages/screens.
- The screens which are star marked are the parent screens i.e. they allow the user to navigate to the child screens. E.g.; Search Type of Proceeding is the parent screen for Add Type of Proceeding and Modify Type of Proceeding, and would allow the user to navigate to the these screens in order to add and modify the proceedings.

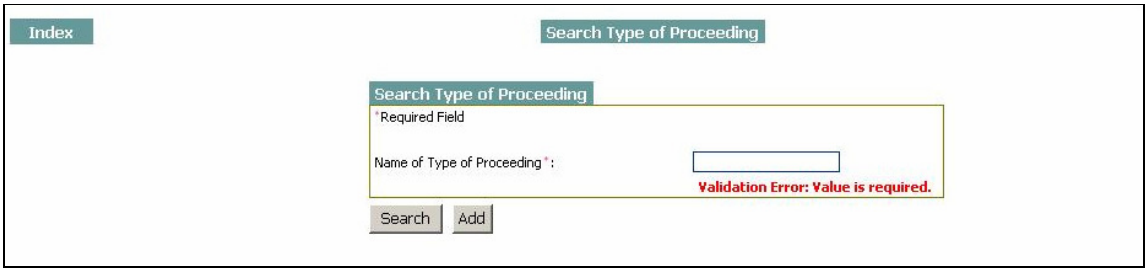


Figure 4.2 Screenshot of Search Type of Proceeding

Screen Name: searchTypeOfProceeding.jsp

Description:

- This screen is displayed when the user clicks on the link Search Type of Proceeding on the screen index.jsp.
- User is displayed a list of proceedings containing the proceeding details after the user clicks the Search button on entering the search criteria.
- If the user clicks on Search button without entering the search criteria, a validation error message is displayed.
- On clicking Modify button, the user navigates to the screen modifyTypeOfProceeding.jsp.
- On clicking Add button, the user navigates to the screen addTypeOfProceeding.jsp.

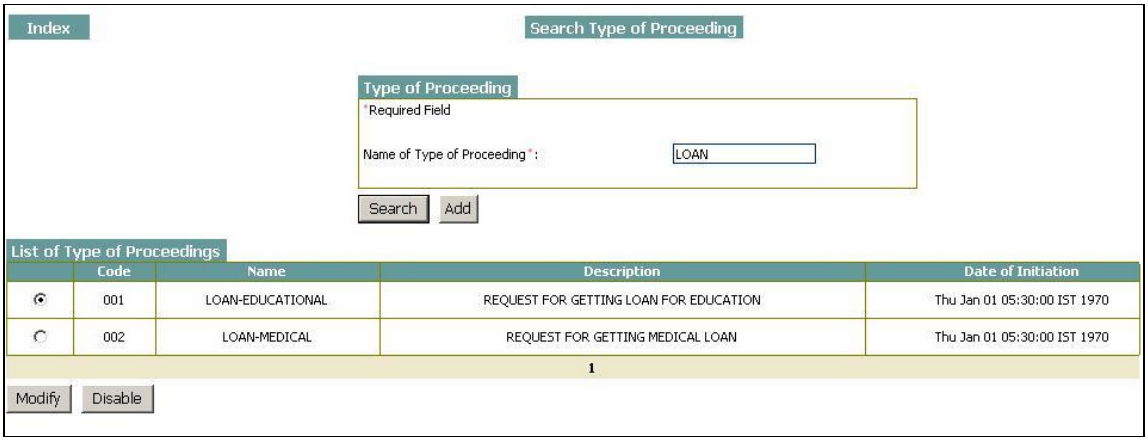


Figure 4.3 Screenshot of Search Type of Proceeding

Screen Name: searchTypeOfProceeding.jsp

Description:

- This screen is displayed when the user clicks on the link Search Type of Proceeding on the screen index.jsp.
- User is displayed a list of proceedings containing the proceeding details after the user clicks the Search button on entering the search criteria.
- User can also modify and disable the selected proceeding. On clicking Modify button, the user navigates to the screen modifyTypeOfProceeding.jsp.
- On clicking the Disable button, the user is displayed a pop up containing YES and NO buttons asking for user confirmation. If the user clicks on YES button, the selected proceeding gets disabled.

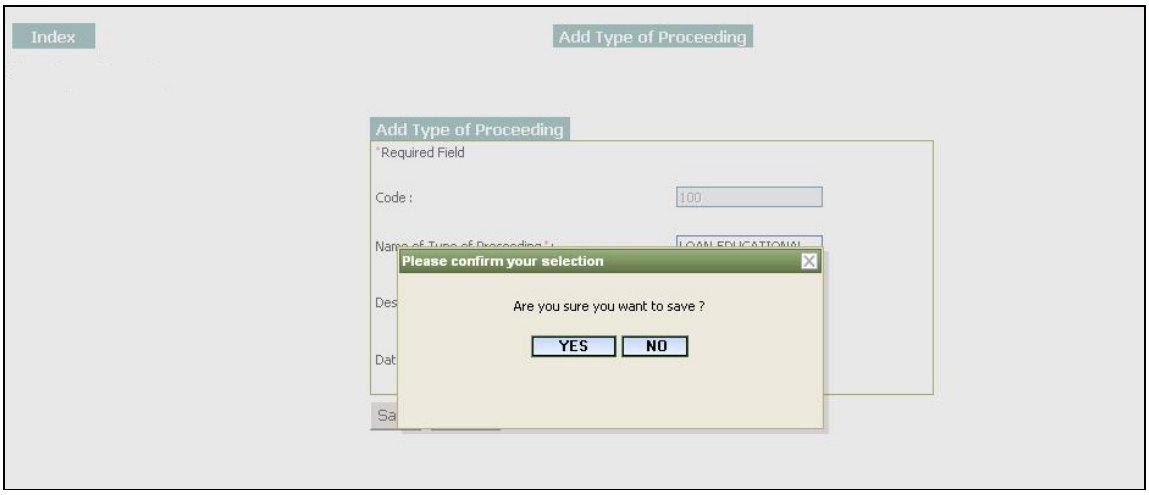


Figure 4.4 Screenshot of Add Type of Proceeding

Screen Name: addTypeOfProceeding.jsp

Description:

- This screen is displayed when the user clicks on Add button on the screen searchTypeOfProceeding.jsp
- User can save a new proceeding by clicking on the Save button.
- User is displayed a pop up containing YES and NO buttons after the user clicks on Save button after entering the mandatory fields. If the user clicks on YES button on the pop up, the record is saved.
- User can return back to the screen – searchTypeOfProceeding.jsp without saving the data by clicking on Return button.
- Code field is an auto generate field.

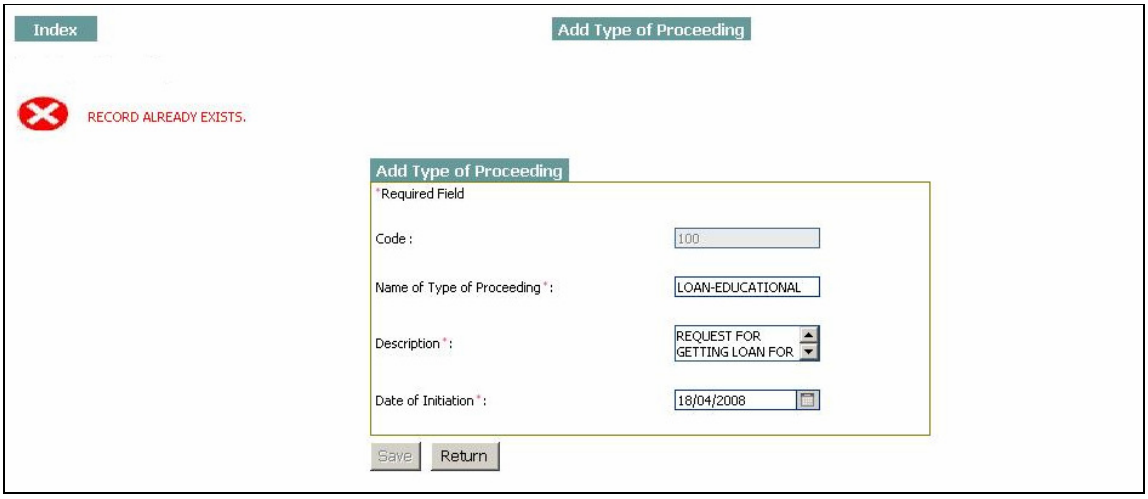


Figure 4.5 Screenshot of Add Type of Proceeding with error msg

Screen Name: addTypeOfProceeding.jsp

Description:

- This screen is displayed when the user clicks on Add button on the screen searchTypeOfProceeding.jsp
- User can save a new task by clicking on the Save button.
- User is displayed a pop up containing YES and NO buttons after the user clicks on Save button after entering the mandatory fields. If the user clicks on YES button on the pop up, the record is saved.
- If the record already exists in the database, user is displayed an error message.
- User can return back to the screen – searchTypeOfProceeding.jsp without saving the data by clicking on Return button.

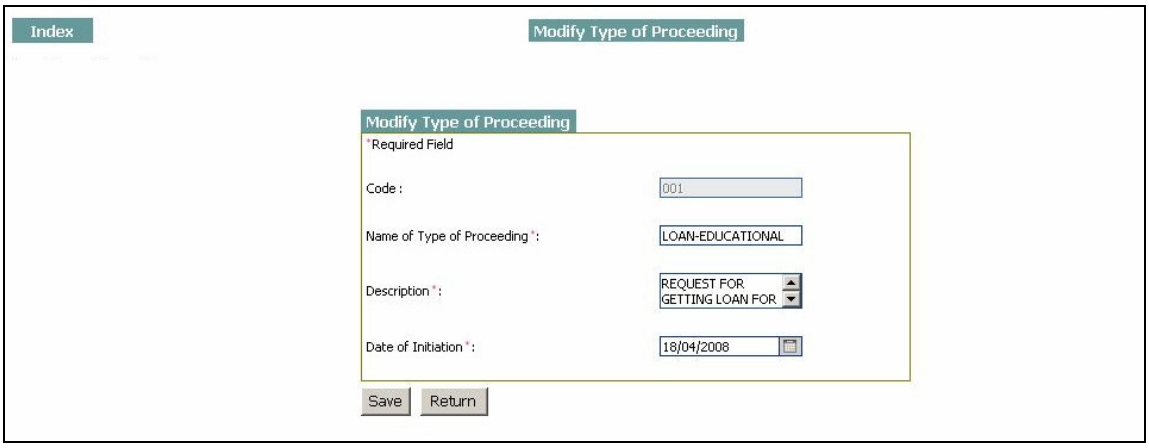


Figure 4.6 Screenshot of Modify Type of Proceeding

Screen Name: modifyTypeOfProceeding.jsp

Description:

- This screen is displayed when the user clicks on Modify button on the screen searchTypeOfProceeding.jsp
- User can modify a proceeding by clicking on the Save button.
- User is displayed a pop up containing YES and NO buttons after the user clicks on Save button after entering the mandatory fields. If the user clicks on YES button on the pop up, the record is modified.
- The values in these fields are filled from the values of the data selected from the list of proceedings displayed on the page searchTypeOfProceeding.jsp
- User can return back to the screen – searchTypeOfProceeding.jsp without saving the data by clicking on Return button.

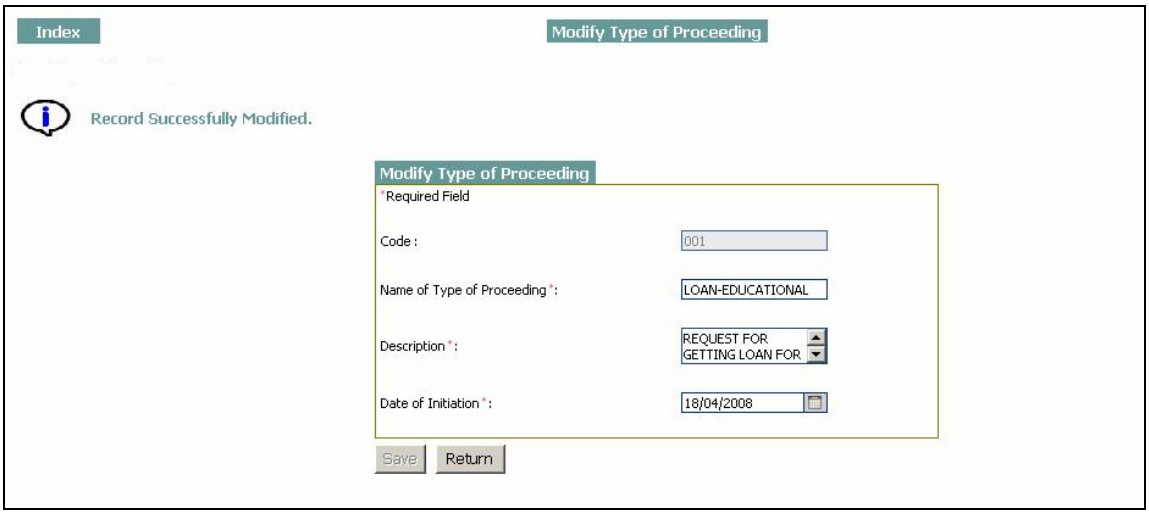


Figure 4.7 Screenshot of Modify Type of Proceeding with msg

Screen Name: modifyTypeOfProceeding.jsp

Description:

- This screen is displayed when the user clicks on Modify button on the screen searchTypeOfProceeding.jsp
- User can modify a proceeding by clicking on the Save button.
- User is displayed a pop up containing YES and NO buttons after the user clicks on Save button after entering the mandatory fields. If the user clicks on YES button on the pop up, the record is modified and a message is displayed.
- The values in these fields are filled from the values of the data selected from the list of proceedings displayed on the page searchTypeOfProceeding.jsp
- User can return back to the screen – searchTypeOfProceeding.jsp without saving the data by clicking on Return button

Table 5.1 UTC – Search Type of Proceeding

Sr. No.	Scenario	Expected Result	Obtained Expected Results (True/False)
1	Successful Pageload.	Displays the screen – searchTypeOfProceeding.jsp with the sections: A) "Search Type of Proceeding" with 1) An enabled textbox "Type of Proceeding". B) Search and Add buttons.	True
2	Enter valid data in the search criteria and click on Search button.	Displays the sections: A) "List of Proceedings" containing the details of those proceedings which match the search criteria B) Modify and Disable buttons.	True
3	Enter invalid data in either of the mandatory fields, say, "Fin@nce" and click on Search button.	Displays an error message - "Only alphanumeric data is allowed."	False
4	Don't enter any data in the search criteria and click on Search button.	Displays an error message - "Value is required."	True
5	Select a row from the list of proceedings displayed and click on Disable button.	Displays a pop up box with buttons YES and NO, asking for user confirmation.	True
5.1	Click on YES button on the pop up.	Displays a message – "Proceeding Successfully disabled."	True
5.2	Click on NO button on the pop up.	Displays the same page again.	True
6	Select a row from the list of proceeding displayed and	Navigates to the screen – modifyTypeOfProceeding.jsp	True

	click on Modify button.		
7	Click on Add button.	Navigates to the screen – addTypeOfProceeding.jsp	True

Table 5.2 UTC – Add Type of Proceeding

Sr. No.	Scenario	Expected Result	Obtained Expected Results (True/False)
1	Successful Pageload.	Displays the screen – addTypeOfProceeding.jsp with the sections A) "Add Type of Proceeding" containing the following input fields: 1) Key – read only. And, Mandatory fields - 2) Type of Proceeding 3) Description 4) Date of Initiation B) Save, and Return buttons.	True
2	Enter valid data in the mandatory fields and click on Save button.	Displays a pop up box with buttons YES and NO, asking for user confirmation.	True
2.1	Click on YES button on the pop up.	Displays a message – "Proceeding Successfully saved." Save button gets disabled.	True
2.2	Click on NO button on the pop up.	Displays the same page again.	True
3	Enter invalid date in "Date of Initiation" field say, "aBcD" and click on Save button.	Displays an error message - "Invalid date format."	True
4	Enter invalid data in either of the mandatory fields, say, "Fin@nce" and click on Save button..	Displays an error message - "Only alphanumeric data is allowed."	False

5	Don't enter any data in the mandatory fields and click on Save button.	Displays an error message - "Value is required."	True
6	Click on Return button.	Navigates to the page searchTypeOfProceeding.jsp.	True

Table 5.3 UTC – Modify Type of Proceeding

Sr. No.	Scenario	Expected Result	Obtained Expected Results (True/False)
1	Successful Pageload.	Displays the screen – modifyTypeOfProceeding.jsp with the sections: A) "Modify Type of Proceeding" containing the following input fields: 1) Key – read only. And, Mandatory fields - 2) Type of Proceeding 3) Description 4) Date of Initiation, B) Save, and Return buttons. These fields contain the data of the row selected from the list of proceedings displayed on searchTypeOfProceeding.jsp	True
2	Enter valid data in the mandatory fields and click on Save button.	Displays a pop up box with buttons YES and NO, asking for user confirmation.	True
2.1	Click on YES button on the pop up.	Displays a message – "Proceeding Successfully modified." Save button gets disabled.	True
2.2	Click on NO button on the pop up.	Displays the same page again.	True
3	Enter invalid date in "Date of Initiation" field say,	Displays an error message - "Invalid date format."	True

	"aBcD" and click on Save button.		
4	Enter invalid data in either of the mandatory fields, say, "Fin@nce" and click on Save button.	Displays an error message - "Only alphanumeric data is allowed."	False
5	Don't enter any data in the mandatory fields and click on Save button.	Displays an error message - "Value is required."	True
6	Click on Return button.	Navigates to the page searchTypeOfProceeding.jsp.	True

6. FUTURE ENHANCEMENT

(1) Enhancement of using automatic DAO generator

Currently the work of DAO generation is done manually which is time consuming and also involves probability of errors in the code. With this proposed enhancement the work of DAO generation will be done by an application – Automatic DAO Generator. It would allow:

- Easier migration of the application from one database implementation to the other.
- Centralizes all Data Access in Separate Layer as all data access operations are now delegated to the DAOs.
- Enables transparency by making the application independent of the persistent storage mechanisms

(2) Enhancement of setting up Help Desk

A help desk facility could be set up to help the users at different locations. The help desk can deal with problems faced by end users in various areas of:

- Operation of the system
- Application Software
- Difficulty in understanding the navigation flow among the GUI.

(3) Enhancement of a task proposal system

Currently the job of breaking down a proceeding to specific tasks is done manually which is time consuming and error prone. The proposed Task proposal system would suggest possible set of tasks for a given proceeding and would allow editing of this proposed set.

(4) Enhancement of Central Authentication Service

CAS is an authentication system originally created by Yale University to provide a trusted way for an application to authenticate a user. **SACR** when integrated with CAS would present the following scenario:

Assume that the user has to access the page http://localhost:8080/MyProject/MyPage.jsp. So, before accessing the page, user would be redirected to:

http://localhost:8080/cas-server-webapp-3.1.1/login?service=
http://localhost:8080/MyProject/MyPage.jsp.

The login service would authenticate the user for that particular service and on completion of authentication; Service Ticket would be generated for accessing that particular service that particular service. Now, the jsp page would be accessed as:-http://localhost:8080/MyProject/MyPage.jsp?ticket=opaque-ticket-string. When the CAS receives a ticket through the validation URL, it checks its internal database to determine whether it saved, in the past, a ticket corresponding to the one it just received. If it did, and if the service associated with that ticket matches the service that was passed in by the application that's requesting validation, it returns the userID associated with that ticket to the requesting application. Otherwise, it refuses to validate the request.

The protocol that the validation URL uses to return data to applications that request validation is straightforward. The CAS responds with two lines (in a text/plain HTTP response); the first line is either yes or no, corresponding to whether the ticket that the application presented is valid or not. If the ticket is valid, the second line contains the userID of the ticket holder - that is, the identification of the user who has authenticated successfully. If the ticket is invalid, the second line is empty.

If the ticket was valid, the CAS immediately removes it so that it cannot be used again. When the cycle is completed, a web application has been able to verify a user without requesting a password.

7. APPENDIX

7.1 DESCRIPTION OF TECHNOLOGY USED

7.1.1 Java Server Faces

7.1.1.1 Introduction

JSF is a development framework that provides a set of standard, reusable GUI components. JSF is used for building Web application interfaces. Java Server Faces defines the following:-

- Component Architecture: It defines a common way to build UI widgets.
- Standard set of UI widgets
- Application infrastructure

JSF has a powerful architecture for displaying components in different ways. It also has extensible facilities for validating input (the length of a field, for example) and converting objects to and from strings for display. JSF can also automatically keep UI components in synch with Java objects that collect user input values and respond to events, which are called *backing beans*. In addition, it has a powerful navigation system and full support for multiple programming languages. These features make up JSF's application infrastructure.

A typical JSF application consists of the following parts [1]:

- JavaBeans components for managing application state and behavior
- Event-driven development (via listeners as in traditional GUI development)
- Pages that represent MVC-style views; pages reference view roots via the JSF component tree.

7.1.1.2 Request Processing Lifecycle [1]

JSF request is processed in six phases, executed in the following order:-

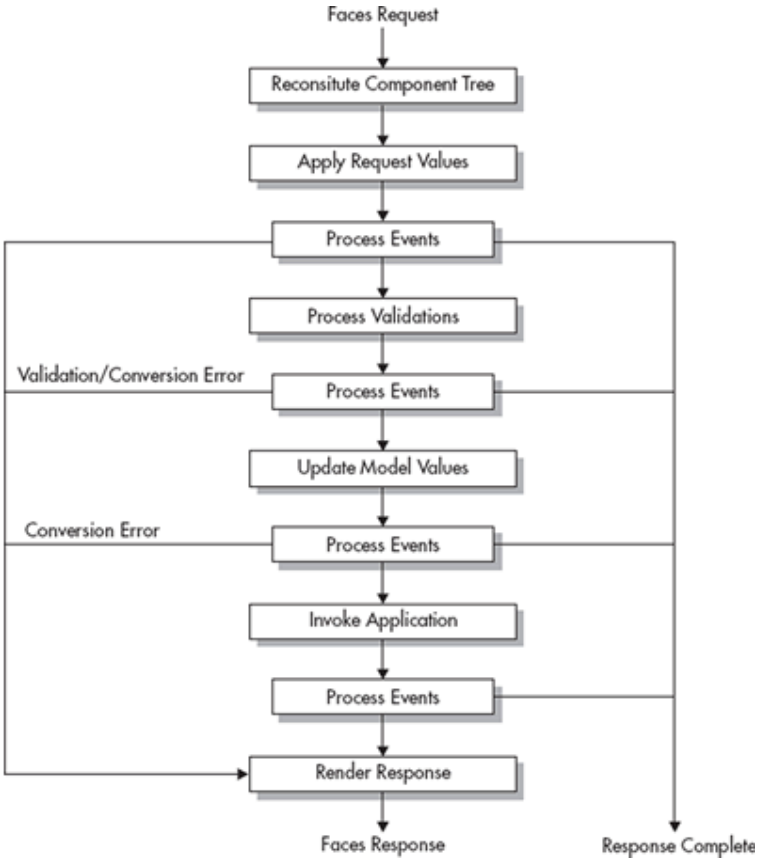


Figure 7.1 JSF Life Cycle

Reconstitute Component Tree

A JSP page in a JSF application is represented by a component tree. This phase starts the request processing by constructing this tree. Each component tree has an identifier that is unique throughout the application. The identifier of a component tree is the path information portion of the request URI. The constructed component tree is then saved in the FacesContext object for processing by the following request processing phases.

Apply Request Values

In this phase, the local value of each component in the component tree is updated from the current request. A value can come from a request parameter, a header, a cookie, and so on. During this phase, a component may queue events. These events will be processed during the process event steps in the request processing lifecycle.

Process Validations

After the local value of each component is updated, in the Process Validations phase, those values are validated if necessary. A component that requires validation must provide implementation of the validation logic. Alternatively, a JSF programmer can register zero or more validators with the component. If one or more external validators are found, the local value of each component will be validated using the validation logic in these external validators.

Update Model Values

This phase can be reached only if the local values of all components in the tree are valid. In this phase, the application's model data is updated. During this phase, a component may again queue events.

Invoke Application

During this phase, the JSF implementation handles any application level events, such as submitting a form or linking to another page.

Render Response

In this phase, the JSF implementation renders the response to the client. The Apply Request Values, Process Validations, Update Model Values, and Invoke Application phases in the request processing lifecycle may queue events in the FacesContext instance associated with the current request. Therefore, the JSF implementation must handle these events after these phases.

7.1.1.3 Working (Control Flow) [7]

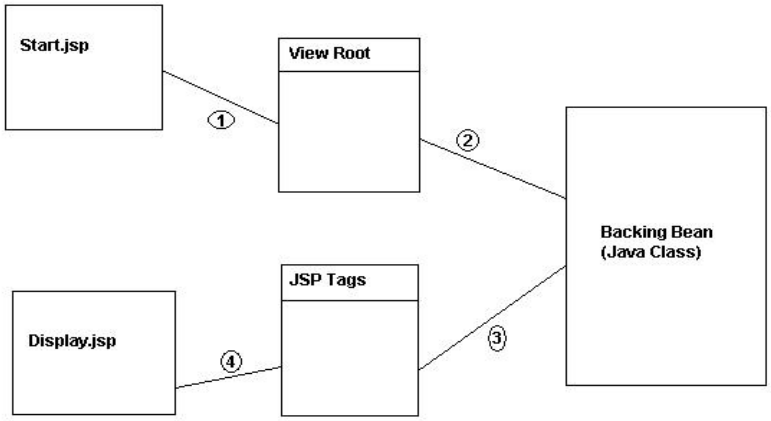


Figure 7.2 Work Flow (JSF Request)

A simple JSF application has the following control flow:-

1. The jsp page is displayed which may contain a number of components. (Here it is start.jsp)
2. The values of the components through the view root reach the backend section.
3. The backend section is comprised of the backing bean which processes the component values.
4. Finally using the navigation rules set in the file faces-config.xml the result page is displayed with the help of jsp tags. (Here it is display.jsp)

7.1.1.4 Advantages

JSF provides the following development advantages:

- Clean separation of behavior and presentation
- Component-level control over statefulness
- Events easily tied to server-side code
- Leverages familiar UI-component and Web-tier concepts
- Offers multiple, standardized vendor implementations

7.1.2 Hibernate

7.1.2.1 Introduction

Hibernate is a Java framework that provides object/relational mapping mechanisms to define how Java objects are stored, updated, deleted, and retrieved. Hibernate helps in developing persistent classes following common Java idiom - including association, inheritance, polymorphism, composition and the Java collections framework. The Hibernate Query Language, designed as a "minimal" object-oriented extension to SQL, provides an elegant bridge between the object and relational worlds. Hibernate is now considered as the most popular object/relational mapping solution for Java.

One of its unique features is that Hibernate does not require the developers to implement proprietary interfaces or extend proprietary base classes in order for classes to be made persistent. Instead, Hibernate relies on Java reflection and runtime augmentation of classes using a powerful, high-performance, code-generation library for Java called CGLIB. CGLIB is used to extend Java classes and implement Java interfaces at runtime.

7.1.2.2 Features of Hibernate

- Hibernate allows automatic generation of primary key. The manner in which the primary key is to be generated is specified in hibernate mapping file.

- Hibernate executes independently of the platform.

- Hibernate helps to minimize the code length. Consider that it is required to update the data in the database. Following JDBC, we would be required to specify an update query after obtaining the connection and statement object. Whereas, Hibernate only requires the user to modify the properties in the POJO class and it checks, if there is any change in the properties of persistent object(s), an appropriate query is generated and fired.

7.1.2.3 Architecture Of Hibernate[3]

It involves the following components:-

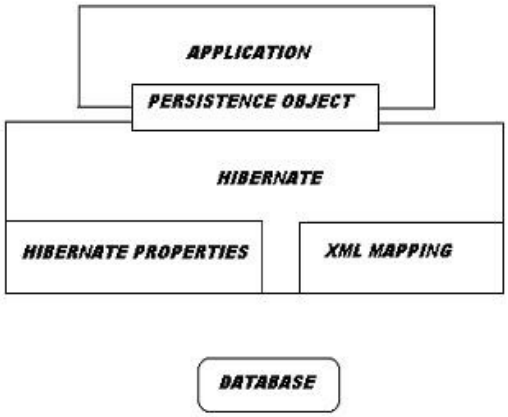


Figure 7.3 Hibernate Architecture

This diagram shows Hibernate using the database and configuration data to provide persistence services (and persistent objects) to the application.

The Hibernate interfaces may be approximately classified as follows:

- Interfaces called by applications to perform basic CRUD and querying operations: These interfaces are the main point of dependency of application business/control logic on Hibernate. They include Session, Transaction, and Query.
- Interfaces called by application infrastructure code to configure Hibernate, most importantly the Configuration class.
- *Callback* interfaces that allow the application to react to events occurring inside Hibernate, such as Interceptor, Lifecycle, and Validatable.
- Interfaces that allow extension of Hibernate's powerful mapping functionality, such as UserType, CompositeUserType, and IdentifierGenerator.

These interfaces are implemented by application infrastructure code (if necessary). JDBC provides a rudimentary level of abstraction of functionality common to relational databases, allowing almost any database with a JDBC driver to be supported by Hibernate.

JNDI and JTA allow Hibernate to be integrated with J2EE application servers.

7.1.2.4 Interfaces [4]

The five core interfaces are used in just about every Hibernate application. Using these interfaces, one can store and retrieve persistent objects and control transactions.

Session interface

The Session interface is the primary interface used by Hibernate applications. An instance of Session is lightweight and is inexpensive to create and destroy. This is important because an application will need to create and destroy sessions all the time, perhaps on every request. Hibernate sessions are *not* threadsafe and should by design be used by only one thread at a time. It may be easier to think of a session as a cache or collection of loaded objects relating to a single unit of work.

SessionFactory interface

The application obtains Session instances from a SessionFactory. The SessionFactory is certainly not lightweight! It's intended to be shared among many application threads. There is typically a single SessionFactory for the whole application—created during application initialization, The SessionFactory caches generated SQL statements and other mapping metadata that Hibernate uses at runtime. It also holds cached data that has been read in one unit of work and may be reused in a future unit of work.

Configuration interface

The Configuration object is used to configure and bootstrap Hibernate. The application uses a Configuration instance to specify the location of mapping documents and Hibernate-specific properties and then create the SessionFactory.

Transaction interface

The Transaction interface is an optional API. Hibernate applications may choose not to use this interface, instead managing transactions in their own infrastructure code. A Transaction abstracts application code from the underlying transaction implementation—which might be a JDBC transaction, a JTA

UserTransaction, or even a Common Object Request Broker Architecture (CORBA) transaction— allowing the application to control transaction boundaries via a consistent API. This helps to keep Hibernate applications portable between different kinds of execution environments and containers.

Query interface

The Query interface allows one to perform queries against the database and control how the query is executed. Queries are written in HQL or in the native SQL dialect of the database. A Query instance is used to bind query parameters, limit the number of results returned by the query, and finally to execute the query.

7.1.2.5 Hibernate Over JDBC

- Java is an object-oriented language. But, to query a database, JDBC is used which is highly relational in nature i.e. to specify a query, it is essential to specify the name of the table to be used (in a string) which essentially turns out to be a relational approach. Hence, it violates the Object Oriented Concepts.
- The tabular representation of data is very much different from the objects used in object oriented applications. This is called Object/relational paradigm. Hence, to follow the basic concepts of object oriented approach, ORM is used.
- ORM stands for Object Relational Mapping, which is process of persisting objects in a relational database. Hibernate works by persisting and restoring plain old Java Objects (POJOs) using a very transparent and low-profile programming model. This task of persisting Java objects to a relational database involves serializing Java objects to a tabular-structured database and vice versa. Essential to this effort is the need to map Java objects to database columns and records in a manner optimized for speed and efficiency.

7.1.3 Spring

Spring is an open source framework, created by Rod Johnson to address the complexity of enterprise application development. Spring makes it possible to use plain JavaBeans to achieve things that were previously only possible with EJBs. However, Spring's usefulness isn't limited to server-side development. Any Java application can benefit from Spring in terms of simplicity, testability, and loose coupling. Spring does many things, but when we strip it down to its base parts, Spring is a lightweight dependency injection and aspect-oriented container and framework.

7.1.3.1 Features of Spring [3]**:**

- **Lightweight**—Spring is lightweight in terms of both size and overhead. The bulk of the Spring Framework can be distributed in a single JAR file that weighs in at just over 2.5 MB. And the processing overhead required by Spring is negligible. Spring is nonintrusive: objects in a Spring-enabled application often have no dependencies on Spring-specific classes.
- **Dependency Injection**—Spring promotes loose coupling through a technique known as dependency injection (DI). When DI is applied, objects are passively given their dependencies instead of creating or looking for dependent objects for themselves. We can think of DI as JNDI in reverse—instead of an object looking up dependencies from a container, the container gives the dependencies to the object at instantiation without waiting to be asked.
- **Aspect-oriented**—Spring comes with rich support for aspect-oriented programming (AOP) that enables cohesive development by separating application business logic from system services (such as auditing and transaction management). Application objects do what they're supposed to do—perform business logic—and nothing more. They are not responsible for (or even aware of) other system concerns, such as logging or transactional support.
- **Container**—Spring is a container in the sense that it contains and manages the lifecycle and configuration of application objects. In Spring, we can declare how each of our application objects should be created,

how they should be configured, and how they should be associated with each other.

- **Framework**—Spring makes it possible to configure and compose complex applications from simpler components. In Spring, application objects are composed declaratively, typically in an XML file. Spring also provides much infrastructure functionality (transaction management, persistence framework integration, etc.), leaving the development of application logic to us.

To restate, when we strip Spring down to its base parts, what we get is a framework that helps us develop loosely coupled application code. Even if that were all that Spring did, the benefits of loose coupling (maintainability and testability) would make Spring a worthwhile framework to build applications on. But Spring is more. The Spring Framework comes with several modules that build on the foundation of DI and AOP to create a feature-filled platform on which to build applications.

7.1.3.2 The Spring Framework [5]

The Spring Framework is made up of several well-defined modules. When taken as a whole, these modules give us everything we need to develop enterprise-ready applications.

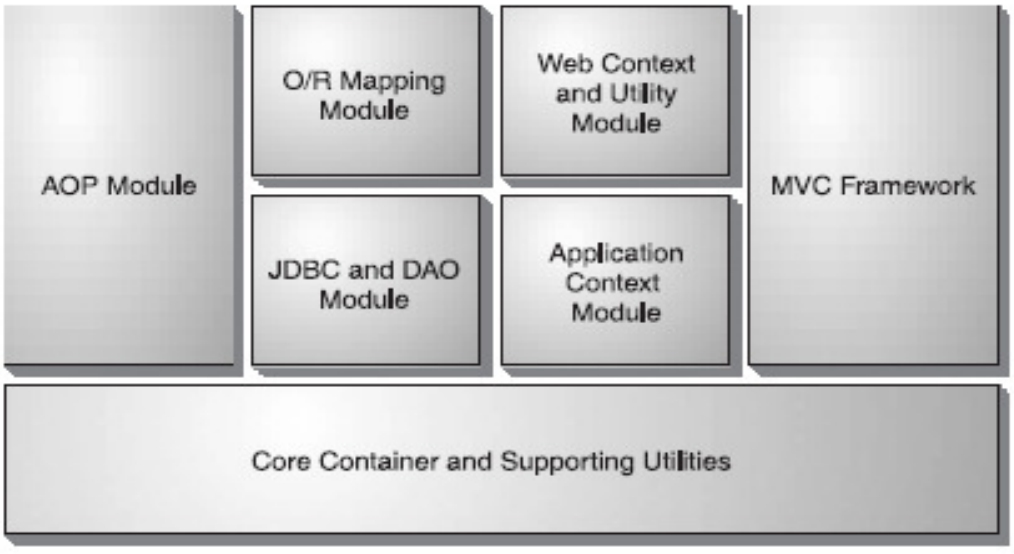


Figure 7.4 Spring Architecture

The various Spring modules are:

The core container

Spring's core container provides the fundamental functionality of the Spring Framework. This module contains the BeanFactory, which is the fundamental Spring container and the basis on which Spring's DI is based.

Application context module

Spring's application context builds on the core container. This module extends the concept of BeanFactory, adding support for internationalization (I18N) messages, application lifecycle events, and validation. In addition, this module supplies many enterprise services such as email, JNDI access, EJB integration, remoting and scheduling.

Spring's AOP module

Spring provides rich support for aspect-oriented programming in its AOP module. This module serves as the basis for developing our own aspects for our Spring enabled application.

JDBC abstraction and the DAO module

Working with JDBC often results in a lot of boilerplate code that gets a connection, creates a statement, processes a result set, and then closes the connection. Spring's JDBC and Data Access Objects (DAO) module abstracts away the boilerplate code so that we can keep our database code clean and simple, and prevents problems that result from a failure to close database resources. This module also builds a layer of meaningful exceptions on top of the error messages given by several database servers.

Object-relational mapping (ORM) integration module

For those who prefer using an object-relational mapping (ORM) tool over straight JDBC, Spring provides the ORM module. Spring's ORM support builds on the DAO support, providing a convenient way to build DAOs for several ORM solutions. Spring doesn't attempt to implement its own ORM solution, but does provide

hooks into several popular ORM frameworks, including Hibernate and Java Persistence APIs.

Java EE Connector API (JCA)

The Java EE Connection API (better known as JCA) provides a standard way of integrating Java applications with a variety of enterprise information systems, including mainframes and databases.

7.2 DESCRIPTION OF TOOLS USED

7.2.1 IBM WebSphere Integration Developer

WebSphere Integration Developer is the "one tool, one set of skills" solution for end-to-end integration in your Service Oriented Architecture. WebSphere Integration Developer is the Eclipse-based tool for building SOA-based Business Process Management and integration solutions across the WebSphere Process Server.

Module and WPS

A module is basically a project where you store all your artifacts (Objects).
When you are creating a module in the tooling, you are actually creating a Java 2 Platform, Enterprise Edition (J2EE) application. A module is packaged and deployed on WebSphere Process Server as an EAR file.

BPEL, Business Processes and Business Rules

Business Process Execution Language *for Web Services* is a language that helps to automate business processes. BPEL processes as well as other components are described by a **WSDL** interface.

A business process is associated with a business rule and this association is depicted in an association diagram. E.g.

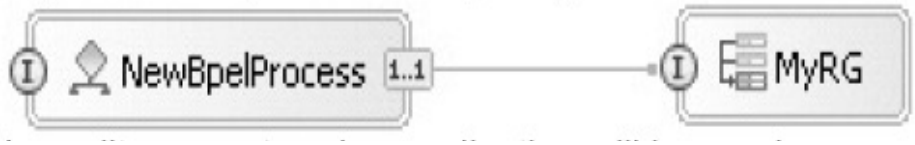


Figure 7.5 Association Diagram

In the diagram 7.2.1.1 NewBpelProcess is wired to a business rule component MyRG. This means that the business process is invoking the business rule, and that the NewBpelProcess component has a reference to the MyRG component.

WebSphere Integration Developer delivers a flexible user interface that allows a client to generate and customize user interfaces that can be used for human interactions within a business process. This comprehensive support includes wizards for generation JavaServer Faces (JSF) based, portlet-based, and forms-based user interfaces.

Some useful features of IBM Websphere Integration Developer:

- Simplifies integration with rich features that accelerate the adoption of service-oriented architecture by rendering existing IT assets as service components, encouraging reuse and efficiency.
- Enables integration developers to assemble complex business solutions -- processes, mediations, adapters, or code components -- requiring minimal skills.
- Enables construction of process and integration solutions using drag-and-drop technology without having a working knowledge of Java
- Enables rapid assembly of business solutions by wiring reusable service components.
- Integrates testing, debugging, and deployment for solution development.
- Enables Business-Driven Development, fully integrating with WebSphere Business Modeler to import models for rapid implementation.
- Operating systems supported: Linux, Windows.

7.2.2 PL/SQL Developer

PL/SQL Developer is an Integrated Development Environment that is specifically targeted at the development of stored program units for Oracle Databases.

Over time more and more business logic and application logic has moved into the Oracle Server, so that PL/SQL programming has become a significant part of

the total development process. PL/SQL Developer focuses on ease of use, code quality and productivity, key advantages during Oracle application development.

Following are the major features of PL/SQL Developer:-

SQL Window

The SQL Window allows you to enter any SQL statement or multiple statements and view or edit the results in a grid. The result grid supports a Query By Example mode to search specific records in a result set. You can easily recall previously executed SQL statements from a history buffer. The SQL editor provides the same powerful features as the PL/SQL Editor.

Powerful PL/SQL Editor

With its Syntax Highlighting, SQL and PL/SQL help, Object Descriptions, Code Assistant, Compiler Hints, Refactoring, PL/SQL Beautifier, Code Contents, Code Hierarchy, Code Folding, Hyperlink Navigation, Macro Library and many other sophisticated features, the Unicode compliant editor appeals to the most demanding user.

Object Browser

This configurable tree-view displays all information that is relevant to PL/SQL development. Use it to get table descriptions, to view object definitions, to create test scripts for the debugger, to enable and disable triggers and constraints, etc.

Integrated Debugger

The integrated debugger features like Run time Exception, Breakpoints, etc.

Query Builder

The graphical Query Builder makes it easy to create new select statements or modify existing ones. Just drag and drop tables and views, select columns for the field list, where clause and order by clause, join tables based on foreign key constraint definitions, and you're done.

7.2.3 Crystal Report Designer

Crystal reports is a business intelligence application meant for designing reports. Crystal Reports is designed to produce the report you want from virtually any data source. The flexibility of Crystal Reports doesn't end with creating reports — the reports can be published in a variety of formats including Microsoft Word and Excel, PDF and even over the Web.

Features provided by Crystal Reports X1:

- **Multiple export formats.** It provides end users with reports in the format they prefer, exports reports to popular formats, including Excel, PDF, XML, HTML, RTF, and more
- **Parameters.** End users can select predefined parameters in a single report to receive a subset of customized data.
- **Alerting.** Highlight specific information for the end users with report alerts -- user-defined messages that appear when certain conditions are met by the data in a report.
- **Powerful formula language.** Crystal Reports allows the use of formulas to carry out various tasks which include, calculating the sum, average, maximum, minimum etc.
- **Customizable templates.** Crystal Reports allows the design and application of customized templates that specify standards in formatting and logic -- including data access operations -- to ensure design consistency across reports. Existing reports can be used as templates.

Stages involved in designing a report

Designing a report involves various phases:

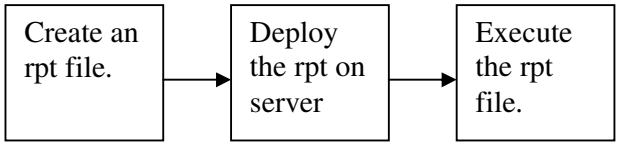


Figure 7.6 Report Generation Phases

Creation of an rpt file involves designing the report using **Crystal Reports X1** and **Eclipse.** It also involves developing a VO which mainly includes a set of properties and some getter and setter methods to access these properties. These properties are then bound to the details section of a report. Ultimately, the purpose of these properties is to hold the data from database. After an rpt file along with VO is created, it is deployed on server and executed.

7.2.4 Sybase PowerDesigner

This tool has been used to support the design phase of the SDLC and hence in generation of the various design documents.

PowerDesigner Key Features

PowerDesigner supports the following modeling techniques

Data Modeling: PowerDesigner supports Conceptual, Logical and Physical Data models, including Warehouse Modeling extensions.

Application Modeling: PowerDesigner supports all UML diagrams and offers advanced object/relational mapping for persistence implementation management. PowerDesigner also supports XML-specific modeling techniques linked to UML and Data models

Business Process Modeling: PowerDesigner supports non-technical business process description as well as process execution language modeling for BPEL4.

PowerDesigner offers specific modeling techniques for specific individuals:

DBAs and developers can take advantage of technology-centric meta-data management capabilities while IT managers and non-technical folks can benefit from business-centric techniques: all integrated into one common, easy to use, framework.

Open support for all major development platforms:

PowerDesigner supports a large number of leading application development platforms like Java J2EE, Microsoft .NET, Web Services and process execution languages like BPEL4.

*Provides an easy to use GUI***:**

PowerDesigner's highly customizable user interface makes common tasks easy while empowering advanced users rapid access to all the features.

7.2.5 Oracle 10g

Oracle is the most widely used database in the world. It runs on virtually every kind of computer. It functions virtually identically on all these machines, which make the Oracle skills and knowledge very portable. Oracle is the most stable Relational Database Management System.
SACR uses Oracle10g DBMS as its backend. The 'g' in Oracle 10g stands for "Grid Computing". Oracle Corporation marks Oracle 10g as "the database for Grid Computing".
The Oracle Server provides efficient and effective solutions with the following features:

Table 7.1 Oracle 10g Features

Feature	Description
Client/Server (distributed processing) environments	To take full advantage of a given computer system or network, Oracle allows processing to be split between the database server and the client application programs.
Large databases and spaces management	Oracle supports the largest of databases, potentially 100 terabytes in size. To make efficient use of expensive hardware devices, it allows full control of space usage.
Many concurrent database users	Oracle supports large number of

	concurrent users executing a variety of database applications operating on the same data. It minimizes data contention and guarantees data concurrency.
High availability	Oracle can work 24x7 with no down time for quite a large amount of time.
Portability	Oracle software is ported to work under different operating systems. Applications developed for Oracle can be ported to any operating system with little or no modification.
Compatibility	Oracle software is compatible with industry standards, including most industry standard operating systems.
Connection ability	Oracle software allows different types of computers and operating systems to share information across network.
High transaction processing performance	Oracle maintains the preceding features with a high degree of overall system performance.
Database enforced integrity	Oracle enforces data integrity, "business rules" that dictate the standards for acceptable data. As a result, the costs of coding and managing checks in many database applications eliminated.
Distributed Systems	-For networked, distributed environments, Oracle combines the data physically located on different computers into one logical database that can be accessed by all network users. -Oracle also offers the heterogeneous option that allows users to access data on some non-Oracle databases transparently.

8. SUMMARY AND CONCLUSION

8.1 SUMMARY

Summary of activities carried out during major project training at TCS can be listed as below:

- Initial Training about the technologies and the tools.
- Requirement Analysis of the project.
- Project Design including DB design analysis and GUI related design.
- Project Development (Coding).
- Document (Code, Diagrams, SDD, etc) Management practice through Visual SourceSafe tool.
- Testing of the project.
- Quality Related Work
- Final Documentation.

8.2 CONCLUSION

Throughout the process, I obtained the experience of working in a large organization and it was a great learning experience. I had the privilege of going through the entire software development lifecycle right from requirement gathering phase.

Working with a globally renowned CMM Level 5 company, was a great learning experience to learn their standards and application areas. This company has widened my horizons of knowledge to a very large extent. Till now I was exposed to a small side of practical learning and with this project I have attained a lot of understanding from the practical point of view.

This project will always remain an integral part of my resume and I hope my wonderful experience here would provide a boost to my career as well.

9. BIBLIOGRAPHY AND REFERENCES

9.1 BIBLIOGRAPHY

1. Mann K. D., *Java Server Faces in Action*, Manning Publications Co., 2005.
2. Walls C. and Breidenbach R., *Spring in Action*, Manning Publications Co., 2005
3. Bauer C. and King G., *Hibernate in Action,* Manning Publications Co., 2005
4. *Hibernate Reference Documentation*, www.hibernate.org
5. *Spring Reference Documentation*, www.spring.org

9.2 REFERENCES

6. www.coreservelets.com/JSF-Tutorial
7. http://www-128.ibm.com/developerworks/web/library/j-jsf1

   http://www-128.ibm.com/developerworks/web/library/j-jsf2

   http://www-128.ibm.com/developerworks/web/library/j-jsf3

   http://www-128.ibm.com/developerworks/web/library/j-jsf4
8. http://www-128.ibm.com/developerworks/web/library/wa-spring1
9. http://www-128.ibm.com/developerworks/web/library/wa-spring2
10. http://www.hsqldb.org