2013

CORE E-PORTAL

MEHTAB ALAM

Jamia Hamdard

1. INTRODUCTION

1.1 What is Core ePortal

Marketing is identifying customer needs, wants and desire converting them into products and services. Sale is we already have the product or service we have to sale it.

Sales is a onetime process (means just selling the product to the customer) but marketing is ongoing process (Identifying the customer needs, developing the desired product, sales and then after sales services or customer.

What is SRFQ

- This Core ePortal is for standard business process whose purpose is to invite the different suppliers into bidding process to bid on products through RFQ (request for quotation).
- RFQ involves price per item, quality level per item and other information in payment terms about products.
- RFQ is process through different modules Admin, technical, financial and production.
- Request for quotation is approved or disproved by administrator or it can be approved only for sampling. Admin can increase the date of biding or he can change the product requirements also.
- In technical analysis quality level is checked and rating level is decided accordingly. And if request is for sampling only then technical analysis is to be done accordingly.
- Then cost analysis is to be done in financial analysis that which product is suitable in financial term.

Real Benefits

- Improve Your Team’s Confidence and Competence
- Increase Client Acquisition and Customer Retention
- Produce More Consistent quotation Results
- Increase Profit Margins

1.2 Purpose

Software developers may install (or modify and then install) purchased software or they may write new, custom-designed programs. The choice depends on the cost of each option, the time available to write software, and the available to write software, and the availability of programmers. Typically, in large organizations computer programmers (or combination of system analyst's programmers) are part of the permanent professional staff. In smaller firms, without programmers, outside programming service may be retained on a contractual basis. Programmers are also responsible for documenting the program, providing an explanation of how and why certain procedures are coded in specific ways. Documentation is essential to test the program and carry on maintenance once the application has been installed.

Before starting any coding of any subsystem we were required to carefully study the information and requirements regarding that application. Once the problem was well though off, and ideas were shared/ discussed among other team members about, how exactly the user interface should be and what is the best way to deduce the logic of the application, we are required to write down the flow of logic and other parameters being used or getting affected on a document of Stock Management System called program specifications document (P S D). This P S D has to get approved from, our manager and users to get further suggestions; also this P S D's are used to prepare the documentation of the final system at later stages and help to keep trace of the changes subsequently made in logic.

The whole application is maintained in the menu from so that it remains user friendly and easy to use and less complex. The system security is maintained by means of the starting form, which takes the required user name and the password.

Developing law's of software:

There are following point must be considering at the development phase:

- Performance requirement
- Exceptional handling
- Acceptance criteria
- Design hints and guidelines

2. SYSTEM ANALYSIS AND REQUIREMENTS

2.1 Feasibility Study

The feasibility of a project can be ascertained in terms of technical factors, economic factors, or both. A feasibility study is documented with a report showing all the ramifications of the project. In project finance, the pre-financing work (sometimes referred to as due diligence) is to make sure there is no "dry rot" in the project and to identify project risks to ensure they can be mitigated and managed in addition to ascertaining "debt service" capability

Technical Feasibility:

Technical feasibility refers to the ability of the process to take advantage of the current state of the technology in pursuing further improvement. The technical capability of the personnel as well as the capability of the available technology should be considered. Technology transfer between geographical areas and cultures needs to be analyzed to understand productivity loss (or gain) due to differences (see Cultural Feasibility). Since we are using Java 6, struts 1.2.9 Tomcat 5.5, weblogic 8.1 and so on so technically our project is feasible.

Managerial Feasibility:

Managerial feasibility involves the capability of the infrastructure of a process to achieve and sustain process improvement. Management support, employee involvement, and commitment are key elements required to ascertain managerial feasibility**.**

Economic Feasibility:

This involves the feasibility of the proposed project to generate economic benefits. A benefit-cost analysis and a breakeven analysis are important aspects of evaluating the economic feasibility of new industrial projects. The tangible and intangible aspects of a project should be translated into economic terms to facilitate a consistent basis for evaluation.

Financial Feasibility:

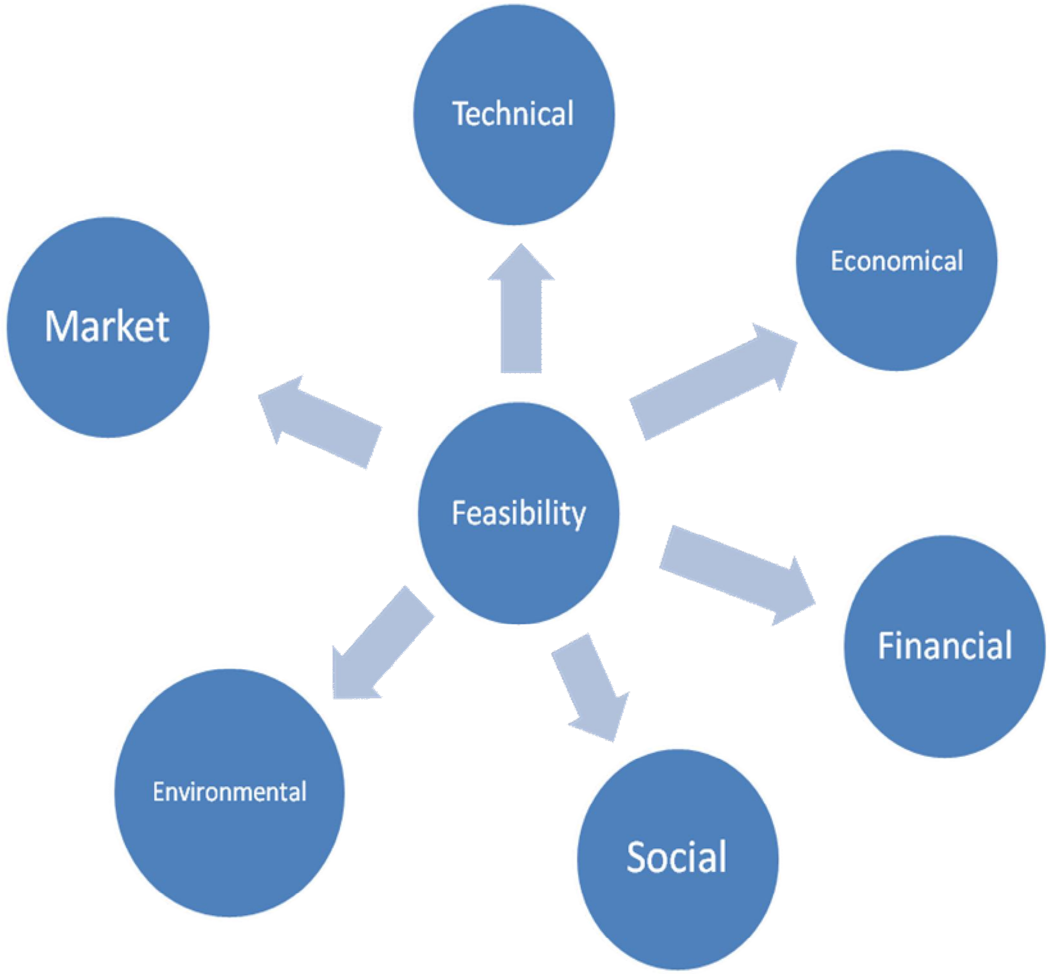
Financial feasibility should be distinguished from economic feasibility. Financial feasibility involves the capability of the project organization to raise the appropriate funds needed to implement the proposed project. Project financing can be a major obstacle in large multi-party projects because of the level of capital required. Loan availability, credit worthiness, equity, and loan schedule are important aspects of financial feasibility analysis.

Environmental Feasibility:

Often a killer of projects through long, drawnout approval processes and outright active opposition by those claiming environmental concerns. This is an aspect worthy of real attention in the very early stages of a project. Concern must be shown and action must be taken to address any and all environmental concerns raised or anticipated. A perfect example was the recent attempt by Disney to build a theme park in Virginia. After a lot of funds and efforts, Disney could not overcome the local opposition to the environmental impact that the Disney project would have on the historic Manassas battleground area.

Market Feasibility:

Another concern is market variability and impact on the project. This area should not be confused with the Economic Feasibility. The market needs analysis to view the potential impacts of market demand, competitive activities, etc. and "divertable" market share available. Price war activities by competitors, whether local, regional, national or international, must also be analyzed for early contingency funding and debt service negotiations during the start-up, ramp-up, and commercial start-up phases of the project.



Feasibility Areas

The Feasibility Study is the preliminary study that determines whether a proposed systems project is technically, financially, and operationally viable. The Alternatives Analysis, usually included as part of the Feasibility Study, identifies viable alternatives for the system design and development. Between them, the documents provide:

- An analysis of the system objectives, functional requirements, and system design concepts.
- A determination of the feasibility of applying automated systems to effectively, efficiently, and economically improve program operations.
- An evaluation of alternative approaches for reasonably achieving the objectives and goals and
- Identification of a proposed approach.

2.1.1 Overview	The Feasibility Study is a critical document which defines the initial system concepts, objectives, requirements, and alternatives. The study also forms the framework for the system development project and establishes a baseline for further studies.

2.1.2 Describe the Status Quo	Following a general overview of the project, the Feasibility Study should establish the "status quo" in the State's management of benefit programs. The current environment may be a manual process, an automated process, or a combination of manual and automated functions. The environment may be paper intensive or dominated by electronic records. The environment may be centralized or distributed. Regardless of attributes, the current operating environment should be described.
	Depending on the systems project being analyzed, the following factors may be addressed:
	Programmatic functions;
	Information architecture;
	System architecture;
	Hardware and software inventory;
	Interface and matching;

	Processing and data flow diagrams;
	Storage and retrieval;
	Inputs;
	Outputs;
	Workload,
	Validation / internal control;
	Security / Privacy;
	Emergency response, back-up, and disaster recovery;
	Personnel; and
	Space and Environment.

2.1.3 Define the Problems	Once the current operating environment has been described, the problems with the current system (previously stated in the Planning APD) should be detailed. Problems may be functional - that is, the system may be incomplete, not fulfilling all the program requirements. Problems may be technical - for example, the system may be too slow, sized too small, or be obsolete and inefficient in terms of hardware or software. Problems may also relate to system cost or to access, limiting the ability of personnel to use system information to full potential. This step should also include a determination of the seriousness of each problem and its effects on factors such as program clients and program financial considerations.
2.1.4 Convert Problems to System Objectives	Once the current operational problems are identified, the State can develop specific system objectives. For example, the system may need to be redesigned to use the powerful attributes of database management software. Or the system may need to be redesigned to provide better service to clients or to support the distributed use and processing of information. Or the system may need to be re-engineered to simplify and streamline work processes for greater efficiency and economy. In defining objectives, various elements must be considered: program needs, costs, level of effort, time schedules, allowable operational changes, ease of future modification and expansion, and system security and reliability. Whatever the element needing improvement, objectives should be defined in a clear, specific, and measurable manner and in

	terms general enough to be met using different automation strategies. System objectives are critical to ensuing analysis - whether conducted to support the Feasibility Study, requirements analysis, or development of testing plans. In terms of the Feasibility Study, the objectives form the framework for the formulation of the initial system requirements, are used to ascertain the acceptability of alternatives, and form the basis for generating costs and benefits during the ensuing Cost/Benefit Analysis. See Table 2-1 on the following page for examples of system objectives.
2.1.5 Identify System Constraints and Assumptions	Constraints are factors that lie outside - but have a direct impact on - the system design effort. Constraints may be: Laws and regulations - for example, State, Federal, or independent regulatory agencies may require specific design approaches for new systems or mandate specific changes to existing systems. Technological - for example, new equipment must be compatible with existing equipment; Socio-political - for example, the Governor mandates that all public assistance ADP functions be combined and managed by a common data base management system; Financial - for example, proposed development and implementation costs must remain within a specified budget. Operational - for example, space, staffing levels, skill mix, and capability and competence factors may limit system options. However, system constraints should not be used to artificially restrict or direct the system. The objective is to plan the best system for the problem to be solved, not to fabricate and impose constraints that limit the system alternatives. As with objectives, system constraints are critical to ensuing phases of the feasibility study. They can affect system requirements and the acceptability of alternatives. Assumptions are factors predicted to apply to the program or systems project. For example, the project's operational or system life - the time required to plan, design, acquire, and implement the system plus its operational life - must be predicted and thus forms a critical assumption during the Feasibility Study. This assumption directly affects the period of time for comparison of costs and benefits of system alternatives and -

	for all practical purposes - sets the range of time within which the system development breakeven point must occur. Four rules apply to making assumptions: Make assumptions when essential information cannot be determined or where the analysis is critically dependent on certain factors, conditions, or future events; State assumptions realistically and in precise terms; Include only assumptions which will affect the analysis; and Document the logic underlying the assumption in the event its soundness needs to be reassessed. In addition to systems life, other common assumptions in cost/benefit analysis are project development and implementation schedule, estimated future workloads, and projected costs and values. Assumptions can be categorized as: Cost/Resource, Functional/Programmatic, Technical and Systems Life.
2.1.6 Develop Initial Functional and Technical Requirements	The Feasibility Study should include an initial statement of the functional and technical requirements for the system. The baseline requirements should relate to the objectives and constraints discussed in the previous sections, summarized as follows: Functional Objectives - the requirements should support mission and program needs. For example, the State may require that the new system improve service to the public and be compatible with and capable of accessing information in related State benefit systems. System Objectives - the requirements should be developed in a manner which will support the objectives. For example, if a system objective is to allow processing at the local level, the initial system requirements should reflect a distributed system and the need to analyze the new information architecture during the system design phase. System Constraints - The functional and technical requirements should conform to, rather than oppose, the system constraints. For example, if the Governor has mandated a single, integrated data base, systems built of separate data bases should not be considered.

	An overview of the system requirements should reflect a broad range of factors, for example: Functional, programmatic requirements; Information needs; System needs; Interface and matching requirements; Processing and data flow needs; Storage and retrieval requirements; Inputs; Outputs; Workload, projected over time; Validation and internal control needs; Security / Privacy requirements; Emergency response, back-up, and disaster recovery; Accessibility requirements for the disabled; and/or Space and Environment. The requirements should be stated briefly and in functional terms, to the extent possible. Their development during the Feasibility Study supports the selection of suitable alternatives. These functional and technical needs are greatly expanded later in the planning phase through the Requirements Analysis.
2.1.7 Assess Project Feasibility	Once the initial system requirements are defined, the State should verify the technical, operational, and financial feasibility of the project. Technical feasibility refers to the capability of current technology and methods of operation in meeting user requirements. Technical feasibility should include consideration of the state of the technology - for example, is the technology "leading edge" (with commensurate risk) or is the technology "mature" (with associated industry standards and lesser risk). Operational feasibility refers to the ability of the enhanced system to fit

	the operational pattern and resources of the organization. Financial feasibility refers to the ability of the State to fund (with Federal financial participation) the costs of developing and implementing the system. Since limited resources - especially human and dollars - may affect feasibility, findings from the technical, operational, and financial feasibility analysis may require redefining or appending the system objectives and constraints.
2.1.8 Identify Alternatives	The first step in identifying alternatives is to survey the possibilities and to consider the wide range of alternatives which may be available. The first part of the process is analytical and judgmental, resulting in eliminating alternatives which are not technically or operationally feasible. Therefore, alternatives are measured against considerations of project feasibility. States should consider more than one technological design alternative when considering an automation project. For example, a system may be centralized, relying on mainframes for the bulk of processing. Or a system may be distributed, relying on personal computers and minicomputers for the bulk of entry and processing. Regardless of technological approach, current systems can frequently be modified - or another State's system may fulfill the programmatic requirements of Federal benefit programs and serve as a transfer model. Whenever possible, several alternatives reflecting different technological approaches - including the options of modifying current systems and transferring another State's system - should be analyzed. The alternatives may represent opposing strategies and should be described in sufficient detail to permit differentiation. All alternatives should meet the established objectives within the system constraints, and depend on costs and benefits to determine the most favorable alternative.
2.1.9 Determine Risks and Effects	For each alternative developed, the effects and risks of the proposed alternative on the current environment should be described: Program impacts - determine how the new system initiative will affect current program operations and new program requirements; Equipment impacts - determine how new equipment requirements will affect current systems and whether technological risks, such as

	obsolescence, maintainability, availability, expandability, reliability, flexibility, and compatibility, are inherent; Software impacts - describe what additions, conversions, or modifications are needed on existing applications and support software; Information impacts - determine how information will be affected, including accessibility, conversion, reformatting into databases, and storage media; Organizational impacts - describe organizational, schedule, accountability, personnel, and skill requirement risks and changes; Operational impacts - set forth the effects on operations, such as user and operating center procedures; user / operator and other relationships; source data processing; data entry procedures; information storage, retention, and retrieval requirements; privacy; output reporting, media, and schedules; system failure and recovery procedures; and security and back-up requirements; Developmental impacts - identify the effect of the development activity on current computing, staffing (including users), space, system security, and contractual support resources; Space and facility impacts - describe the effect on space, both in terms of square footage and necessary modifications to facilities; and Cost impacts - set forth financial risks and factors that may affect developmental or operational costs and influence the development, design, and operation of the proposed system.
2.1.10 Determine Risks and Effects	For each alternative developed, the effects and risks of the proposed alternative on the current environment should be described: Program impacts - determine how the new system initiative will affect current program operations and new program requirements; Equipment impacts - determine how new equipment requirements will affect current systems and whether technological risks, such as obsolescence, maintainability, availability, expandability, reliability, flexibility, and compatibility, are inherent; Software impacts - describe what additions, conversions, or modifications are needed on existing applications and support software; Information impacts - determine how information will be affected, including accessibility, conversion, reformatting into databases, and

	storage media; Organizational impacts - describe organizational, schedule, accountability, personnel, and skill requirement risks and changes; Operational impacts - set forth the effects on operations, such as user and operating center procedures; user / operator and other relationships; source data processing; data entry procedures; information storage, retention, and retrieval requirements; privacy; output reporting, media, and schedules; system failure and recovery procedures; and security and back-up requirements; Developmental impacts - identify the effect of the development activity on current computing, staffing (including users), space, system security, and contractual support resources; Space and facility impacts - describe the effect on space, both in terms of square footage and necessary modifications to facilities; and Cost impacts - set forth financial risks and factors that may affect developmental or operational costs and influence the development, design, and operation of the proposed system.
2.1.11 Rank Alternatives	If more than three or four alternatives have been developed, the State should rank alternatives so that only the most likely to achieve the system objectives efficiently, effectively, and economically are analyzed during the cost/benefit analysis. Criteria for ranking the alternatives should be established and may include factors which: Minimize personnel expenses over the system's operational life; Require minimal physical facility changes; Assure high levels of availability, reliability, maintainability, or expandability; Meet requirements for ease of use and ready access to information; Achieve desired distribution of processing to minimize point-of-entry delays; Achieve redundancy to guard against total system outages; Limit development time; or Retain a centralized information repository for reasons of security. Once the State has isolated no more than four and no less than two

	viable alternatives - one of which is the status quo - the cost/benefit determination may proceed.

Scope of Feasibility Analysis

In general terms, the elements of a feasibility analysis for a project should cover the following:

Need Analysis:

This indicates recognition of a need for the project. The need may affect the organization itself, another organization, the public, or the government. A preliminary study is then conducted to confirm and evaluate the need. A proposal of how the need may be satisfied is then made. Pertinent questions that should be asked include:

Is the need significant enough to justify the proposed project?

Will the need still exist by the time the project is completed?

What are the alternate means of satisfying the need?

What are the economic, social, environmental, and political impacts of the need?

Process Work:

This is the preliminary analysis done to determine what will be required to satisfy the need. The work may be performed by a consultant who is an expert in the project field. The preliminary study often involves system models or prototypes. For technology-oriented projects, artist's conception and scaled-down models may be used for illustrating the general characteristics of a process. A simulation of the proposed system can be carried out to predict the outcome before the actual project starts.

Engineering & Design:

This involves a detailed technical study of te proposed project. Written quotations are obtained from suppliers and subcontractors as needed. Technology capabilities are evaluated as needed. Product design, if needed, should be done at this time.

Cost Estimate:

This involves estimating project cost to an acceptable level of accuracy. Levels of around -5% to +15% are common at this level of a project plan. Both the initial and operating costs are included in the cost estimation. Estimates of capital investment and of recurring and nonrecurring costs should also be contained in the cost estimate document. Sensitivity analysis can be carried out on the estimated cost values to see how sensitive the project plan is to the estimated cost values.

Financial Analysis:

This involves an analysis of the cash flow profile of the project. The analysis should consider rates of return, inflation, sources of capital, payback periods, breakeven point, residual values, and sensitivity. This is a critical analysis since it determines whether or not and when funds will be available to the project. The project cash flow profile helps to support the economic and financial feasibility of the project.

Project Impacts:

This portion of the feasibility study provides an assessment of the impact of the proposed project. Environmental, social, cultural, political, and economic impacts may be some of the factors that will determine how a project is perceived by the public. The value added potential of the project should also be assessed. A value added tax may be assessed based on the price of a product and the cost of the raw material used in making the product. The tax so collected may be viewed as a contribution to government coffers.

Conclusions and Recommendations:

The feasibility study should end with the overall outcome of the project analysis. This may indicate an endorsement or disapproval of the project. Recommendations on what should be done should be included in this section of the feasibility report.

2.2 Requirement Analysis

Requirements analysis is the process of analyzing the information needs of the end users, the organizational environment, and any system presently being used, developing the functional requirements of a system that can meet the needs of the users.

After detailed studying and analysis I found that in the existing system the RFQ records were maintained by the organization manually which led to great deal of problems for the people.

System Analysis Methods

Review the written documents Studying: The existing manuals and forms we gather information regarding the existing system. There are some manuals that describe the role of each person in the share MANAGEMENT department. They keep user and shares record for the concerned persons.

On-site observation: Observation must be carried out to gather the information about the existing system. On visiting the India bulls and Share khan office in Noida the massive information is collected about the existing system. The enormous volume of inefficiency and disorganization was there regarding to users expectations. Each user are dependent on the brokers.

Interview: To know staff or officers perception regarding the manual system an interview is conducted .During the interview we came to know about various steps they follow in handling of shares and records of persons, what are the difficulties they face and other correspondences.

Questionnaires: Questionnaires can be used to get information from large groups of people. Advantages of questionnaires are anonymity and evidence of trends. Various questionnaires are made to the employee or staff in regard to the requirement they need. A set of question are asked to know about various necessity the desire like Hindi Subject data entry, edit the wrong entry made, listing the pending details, security to the records etc.

Analysis Modeling:

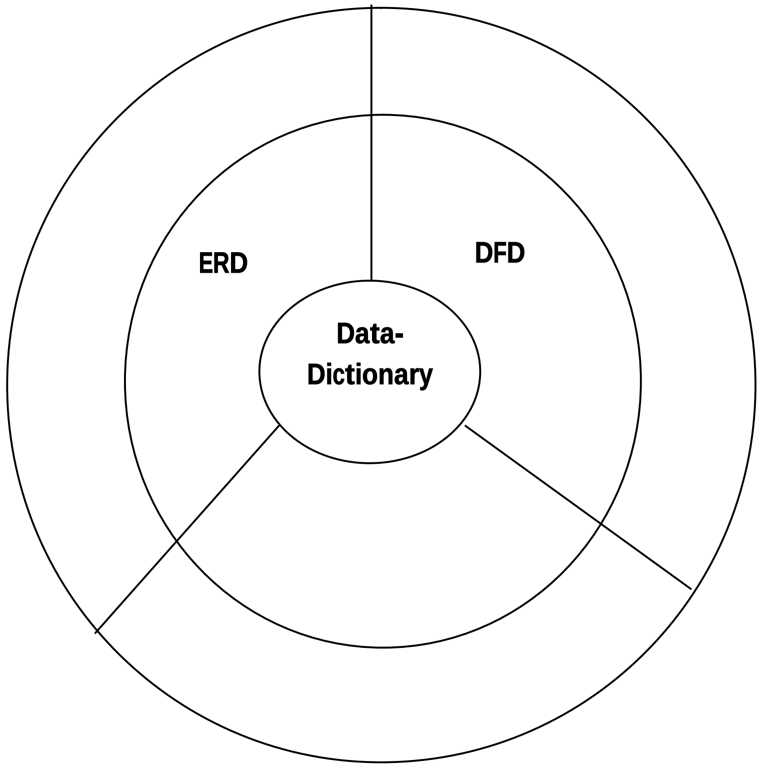
At a technical level software engineering begins with a series of modeling a tasks that lead to a complete specification of requirements and a comprehensive design representation for the software to be built. The analysis model, actually a set of models, is the first technical representation of a system .Over the years many methods have been proposed for the analysis modeling. However, two now dominate. The first, structured analysis is a classical modeling method and another approach, Object-oriented analysis.

The Elements of the Analysis Model:

The analysis model must achieve three primary Objectives:

- To describe what the customer requires,
- To establish a basis for the creation of a software design, and
- To define a set of requirements that can be validated once the software is built.

To accomplish these objectives, the analysis model derived during structured analysis takes the form illustrated in fig below



- ➢ At the core of the model lies the Data Dictionary- a repository that contains descriptions of all data objects consumed or produced by the software. Three different diagrams surround the core. The entity relation diagram (ERD), depicts relationships between data objects. The ERD is the notation that is used to conduct the data modeling activity. The attributes of each data object noted in the ERD can be described using a data object description.
- ➢ The data flow diagram (DFD) serves two purposes:
  - To provide an indication of how data are transformed as they move through the system and

- To depict the functions (and sub functions) that transforms the data flow. The DFD provides additional information that is used during the analysis of the information domain and serves as a basis for the modeling of function. A description of each function presented in the DFD is contained in a process specification (PSPEC).

- The state transition diagram (STD) indicates how the system behaves as a consequence of external events. To accomplish this, the STD serves as the basis for behavioral modeling. Additional information about the control aspects of the software is contained in the control specification.

2.3 System Requirements

An accurate and through understanding of system requirements is essential to the success of any Software Development Process. All further stages of SDLC like system analysis, design and coding depend on how accurate well prepared and thoroughly understood the System Requirements Specification is. Poorly analyzed requirements will disappoint the user no matter how well designed and the well coded the software is.

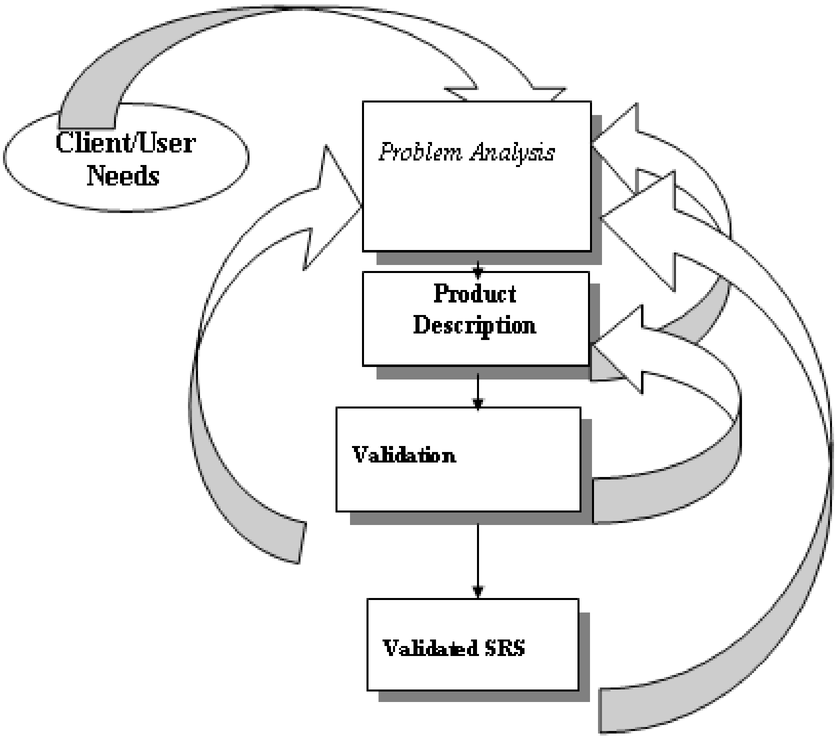
Requirement specification appears to be a relatively simple task but the chances of misinterpretation is very high, ambiguity is probable and communication gap between customer and developer is bound to bring confusions. Requirement Specifications begin with a clear and concise heading stating in a sentence the task to be performed (i.e. work objective). For this, we have to identify the problem first. Problem specifications serve as the basis for identifying work objective that helps in describing the requirements in technical and precise statements. After the initial specification reports are received, they are analyzed and refined through customer developer interaction. System Analysis follows to determine feasibility and Cost Benefit Analysis.
A complete understanding of requirement specification of the new system is very important for the successful development of the software product. Requirement specification is the foundation in the process of software development. All further development like System Analysis, System Design, and Coding will depend on how accurate and well prepared the requirement.

Overall Description

Requirement specification appears to be relatively simple task, but appearances are often deceiving. There is always a chance of wrong specification because of communication gap between user and developer, ambiguity in requirement or a wrongly specified problem. Requirement Specification begins with a clear statement of the problem and the task to be performed. Then requirements are described in a technical manner in precise statements.

After the initial specification reports are received, they are analyzed and refined through user developer interaction. System Analysis follows to determine and cost benefit analysis**.**

An SRS is basically an organization's understanding (in writing) of a customer or potential client's system requirements and dependencies at a particular point in time (usually) prior to any actual design or development work. It's a two-way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at



The SRS document itself states in precise and explicit language those functions and capabilities a software system (i.e., a software application, an ecommerce Web site, and so on) must provide, as well as states any required constraints by which the system must abide. The SRS also functions as a blueprint for completing a project with as little cost growth as possible. The SRS is often referred to as the "parent" document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it.

It's important to note that an SRS contains functional and nonfunctional requirements only; it doesn't offer design suggestions, possible solutions to technology or business issues, or any other information other than what the development team understands the customer's system requirements to be.

2.3.1 General Constraints

The constraints that are imposed on the development of this project are outlined below.

- The system will be accessed through an internet browser (predominantly Microsoft's Internet Explorer). The front end has to therefore be a HTML based screen designer. The proposed system will be using HTML only.
- The data store will be MySQL. This is the data store.
- The data from MySQL will be retrieved using the Java Data Base Connectivity technology, with the usage of the Struts and Hibernate feature of Java.
- JSP technology will be used as the go between Java and HTML. The JSP script will be embedded in HTML.
- The above requirement imposes that a JSP server will be available and running to access the proposed system.
- The security measures should be as outlined before. Only Moderator or administrator should be able to access a particular module.
- The project is to be developed in a modular fashion adhering to standard programming practices.

Assumptions and Dependencies of the system

The system depends on a few parameters that are necessitated by Sierra Atlantic. These dependencies are as outlined below.

- The login will be a part of the existing database. SMS is a portal which enables every employee of SMS to login using his user ID and password. On a successful login the user will be able to access all the resources, this project included.
- The user ID for the session is maintained by SMS. This will be used by the proposed system also for the user ID.
- The proposed system assumes that the maintenance of the database will be done externally to this system. Any changes to the existing table structure will mean that the proposed system's coding might also need to be altered.
- For the purposes of developing this system, a dummy database will be used. This will serve as a place holder for the original database.

2.3.2 Hardware Requirements

Client Side:

Web Browser	Processor	RAM	Disk Space
IE 6.0, Netscape, Firefox	Pentium III at 500 MHz	256 MB	10 GB

Server Side:

Web Server	Processor	RAM	Disk Space
Apache Tomcat 5.5	Pentium IV at 2.67 GHz	2 GB	20 GB

2.3.3 Software Requirements

Platform: Microsoft Windows XP and its latest versions

Primary Interface/Frame (Front End): HTML, XHTML, XML, Ajax, JavaScript

Core Technology: Core Java and J2EE

Core Framework: Struts 1.3.8

Environment: Net Beans IDE 6.7.1, Dreamweaver 8.0

Web Server: Apache Tomcat 6.0.18

Application Server: BEA Web Logic Application Server

Connectivity: Hibernate, JDBC (Java database connectivity)

Secondary Interface (Back End): MySql 5.0

Application Components: Java Beans and Enterprise Java Beans

Core Architecture: MVC (Model-view-controller)

2.3.4 Functional requirements

1. **Configuration**
    - A. Configure Enterprise
        - i. Modify
    - B. Role Manage
        - i. Add
        - ii. Update
    - C. Employee Manage
        - i. Add
        - ii. Update
        - iii. Activate/DeActivate
2. **Security**
    - A. Allocate Functionality to Roles
    - B. Allocate Functionality to Employee
    - C. Change Password
3. **Request Specification**
    - A. Product Requirement Specification by Production
        - i. New Specification
        - ii. Update Specification
        - iii. Remove Specification
        - iv. Active/DeActivate Specification
    - B. Product Requirement Specification Posting for suppliers by Admin
        - i. New Posting
        - ii. Update Posting
        - iii. Remove Posting
        - iv. Active/DeActivate

4. **Configure Supplier**
    - A. Add New Supplier
    - B. Update Supplier
    - C. Activate/DeActivateSupplier
    - D. Remove Supplier
    - E. Add Supplier Contacts
    - F. Update Supplier Contacts
    - G. Remove Supplier Contacts
5. **Quotation Posting**
    - A. Add Quotation
    - B. Upadte Quotation
6. **Technical Analysis**
    - A. Review Technical Specification &Compare
    - B. Technical Feedback on Supplier
    - C. View Supplier’s Feedback History
    - D. Initial Technical Approval or Rejection
7. **Financial Analysis**
    - A. Review Financial Specification &Compare
    - B. Financial Feedback on Supplier
    - C. View Supplier’s Feedback History
    - D. Initial Financial Approval or Rejection
8. **Admin Approval**
    - A. Approval For Order or Rejection
    - B. Approval for Sampling
    - C. View RFQ Status &Compare
9. **Sample Request**
    - A. Sapmle Demand

10. Supplier Sampling Confirmation& Evaluation

A. Sample Confirmation

B. Sample Evaluation

11. Sample Technical Approval

A. Review Sampling Technical Specification & compare

B. Sampling Technical Feedback on Supplier

C. Final Sampling Technical Approval or Rejection

12. Sample Financial Approval

A. Review Sample Financial Specification &Compare

B. Sample Financial Feedback on Supplier

C. Final Sampling Financial Approval or Rejection

13. Sample Operational Approval

A. Review Sample Operational Specification &Compare

B. Sample Operational Feedback on Supplier

C. View Supplier’s Feedback History

D. Final Sample Operational Approval or Rejection

14. Admin Sample Approval

A. View Sample Technical Feedback

B. View Sample Financial Feedback

C. View Sample Operational Feedback

D. Admin Sample Approval &Order

E. Bid Date Manage

2.3.5 Nonfunctional Requirements

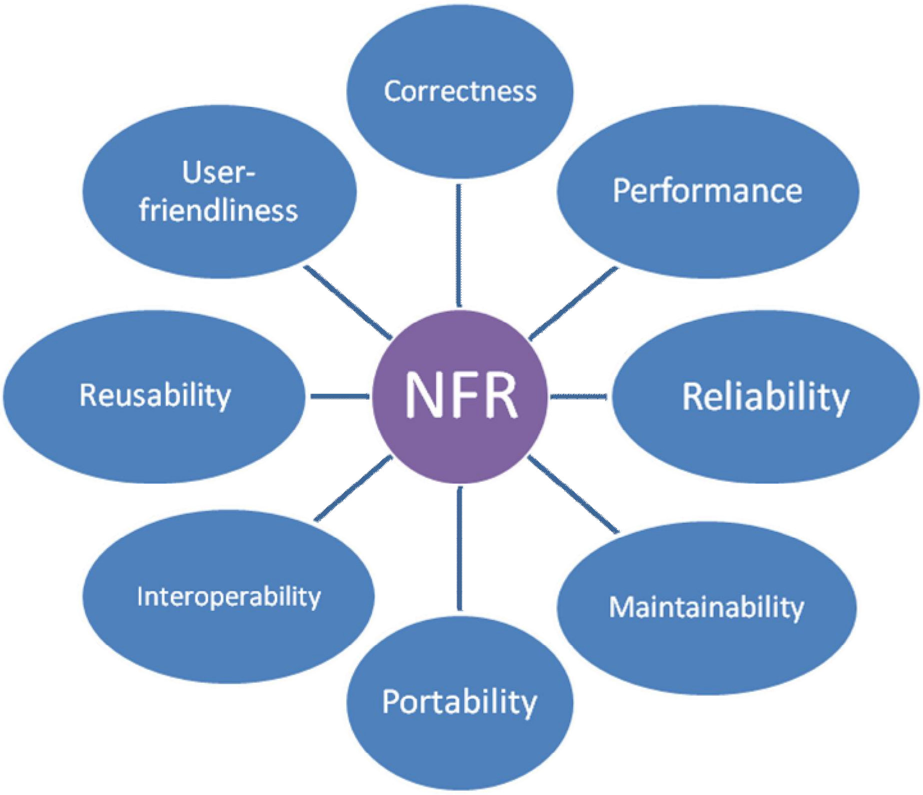


Fig: nonfunctional requirements

Availability: - The system should be properly supported by the required H/W and S/W interface needs so that any user can access the data from any place sitting on the internet connectivity node.

Maintainability: - It is a system which incorporates this feature as it is the system whose defects can be easily removed by the programmers. This system can be easily adapted to meet the changing requirements of its users or the changing environment in which it operates.

Portability: - its design is relatively machine independent. It is being coded in languages which are well-defined & meet to the standards, hence more portable.

Visibility: - Any user can view the menu of the system and can work out the processing even from a remote place. There is no hiding of data and incorporates easy availability and accessibility.

Efficiency: - It is an efficient system because it fulfills most of the requirements being mentioned by the users. It uses least amount of devices & memory space for execution of the package on the systems at the client's end.

Reliability: - It is a very reliable system. Two separate processes of the package are unique & non-interdependent to each other. The execution of one process does not effect the processing of second one. The complete processing does not effect the normal operations of the company terminals.

Safety & Security: - The processes/operations under each consideration are completely equipped with alert messages as and when required. The software never permits any mishandling or misfiring of data in the database tables at any point of processing.

Standard Compliance: - It follows a standard format or layout in the designing as well as operational phases. This helps the user to work on the system in an effective and efficient way at any site of the office. They can work on its database to keep all the concurrent updating.

2.3.6 Performance Requirements

The following performance characteristics were taken care of in developing the systems:

User Friendliness:

The system is easy to learn and understand. A naive user can also use the system effectively, without any difficulty.

User satisfaction:

The system is such that it stands unto the user's expectation.

Response time:

The response time of all the operations is very low. This has been made possible by careful programming.

Error handling:

Response to user errors and undesired situations has been taken care of to ensure that the system operations without halting in case of such situation and proper are given to user.

Safety:

The program is able to avoid catastrophic behavior.

Robustness:

The system recovers from undesired events without human intervention.

Security:

This system provides protection of information through the mechanism of password which is incorporated in it. Therefore only authorized people can access the Databases to validations. This results in a thorough testing of the details, and the system is such that likely to change and modifications can be easily incorporated in it.

Portability:

The system can move to a new hardware/ operating system after making minor modifications to it. It’s ensured that the system does not halt in case of undesired situation or events exception conditions are taken care of providing the corresponding exception responses while developing the system.

Components of System

Based on the analysis done, the system was partitioned into various functional modules to bring down the complexity of the system. These modules help in identifying specific functions and also create an easy to understand and develop approach during system development.

- Requirement posted by the Organization
- Configure new Suppliers
- Quotation posted by the Suppliers
- Quotation approved/rejected by Admin

- Request for the samples
- Samples posting by Supplier
- Samples approved by organization
- Final Approval

2.3.7 Development Environment Requirements

IDE	:	NetBeans IDE 6.5
Processor/RAM/HDD	:	Intel Pentium 4/1 GB Ram/80 GB HDD

2.4 Technology Used

Introduction

The technologies used at the back-end and front-end are discussed here:

2.4.1 JAVA

History of JAVA

Java language was developed by James Gosling and his team at sun micro systems and released formally in 1995. Its former name is oak. Java Development Kit 1.0 was released in 1996. To popularize java and is freely available on Internet.

Overview of Java

Java is loosely based on C++ syntax, and is meant to be Object-Oriented Structure of java is midway between an interpreted and a compiled language. The java compiler into Byte Codes, which are secure and portable across different platforms, compiles Java programs. These byte codes are essentially instructions encapsulated in single type, to what is known as a java virtual machine (JVM), which resides in standard browser.

JVM verifies these byte codes when downloaded by the browser for integrity. JVM is available for almost all OS. JVM converts these byte codes into machine specific instructions at runtime.

Features of JAVA

Java is object-oriented language and supports encapsulation, inheritance, polymorphism and dynamic binding, but does not support multiple inheritances. Everything in java is an object except some primitive data types.

Java is portable architecture neutral that is java programs once compiled can be executed on any machine that is enabled.

JAVA is distributed in its approach and used for Internet programming.

Java is robust, secured, high performing and dynamic in nature.

Java supports multithreading. Therefore different parts of the program can be executed at the same time.

2.4.2 JSP

Overview of JSP Technology

The JSP technology will be used to interface HTML with Java. The JSP technology provides a seamless connection with Java and presents an easy to use, Java-like programming constructs that can be scripted within HTML files. Java Server Pages is a technology for developing web pages that include dynamic content.

A JSP page contains standard markup language elements, such as HTML tags, just like a regular web page. A JSP page also contains special JSP elements that allow the server to insert dynamic content in the web page.

Advantages of JSP

- JSP supports both scripting and element-based dynamic content.
- Allows developing custom tag libraries.
- JSP pages are precompiled for efficient server processing.
- JSP pages can be used in combination with servlets that handle the business logic.
- High Security.
- Can run on any J2EE compatible web Server.
- It can run on any OS that have J2EE compatible web server.
- JSP separates the dynamic and static parts.
- High Quality tool supports.
- JSP supports N tier Application.
- Write Once, Run Everywhere.
- JSP is vender Neutral.

2.4.3 HTML

HTML (hyper text markup language) is a language used to create hypertext documents that have hyper links embedded in them. It consists of tags embedded in the text of a document. With HTML we can build web pages or web document s. it is basically a formatting language and not a programming language. HTML is used to design the interface to the application. HTML is platform independent.

Examples of browsers used to be web pages include:

- Firefox
- Netscape
- Internet Explorer

2.4.4 INTERNET EXPLORER 5.0

An Internet browser is required to access the training function automation system. IE 5.0 is the most widely used browser at Sierra Atlantic. Hence the system was designed for IE; it will work equally efficiently with any other browser.

2.4.5 JAVA SCRIPT

Java script is a general purpose, prototype based, object oriented scripting language developed jointly by sun and Netscape and is meant for the WWW. Java script borrows most of its syntax from java but also inherits from awk and perl, with some indirect influence from self in its object prototype system.

Java Script is almost as easy to learn as HTML and it can be included directly in HTML documents. Java Script was developed independently of java. Java script is a high level scripting language that does not depend on or expose particular machine representations or operating system services.

Features:

- Java script is embedded into HTML documents and is executed with in them.
- Java script is browser dependent.
- JavaScript is an interpreted language that can be interpreted by the browser at run time.
- Java script is loosely typed language
- Java script is an object-based language.
- Java script is an Event-Driven language and supports event handlers to specify the functionality of a button.

2.4.6 JSTL(Java Server Pages Standard Tag Library)

The Java Server Pages Standard Tag Library (JSTL) encapsulate, as simple tags , core functionality common to many JSP applications. For example, instead of suggesting that you iterate over lists using a scriptlet or different iteration tags from numerous vendors, JSTL defines a standard tag that works the same everywhere. This standardization lets you learn a single tag and use it on multiple JSP contains. Also, when tags are standard, contains can recognize the and optimize their implementations.

JSTL has support for common, structural tasks such as iteration and conditionals, tags for manipulation XML documents, internationalization and locale-sensitive formatting tags, and SQL tags. It also introduces a new expression language to simplify page development, and it provides an API for developers to simplify the configuration of JSTL tags and the development of custom tags that conform to JSTL conventions.

JSTL (Java Server Pages Standard Tag Libraries) is a collection of JSP custom tags developed by Java Community Process, www.jcp.org. The reference implementation is developed by Jakarta project, Jakarta.apache.org.

The latest version of JSTL is JSTL1.1, which require a JSP container that supports the Java Servlet 2.4 and Java Server Pages 2.0 specification. Jakarta Tomcat 5 supports these specifications.

The previous version of JSTL is JSTL1.0, which require a JSP container that supports the Java Servlet 2.3 and Java Server Pages 1.2 specification. Jakarta Tomcat 4 supports these specifications.

JSTL offers tags through four libraries:

- Core-Basic scripting functions.
- Xml-XML procession.
- Fmt-Internationalization of formation
- Sql-Data base accessing.

2.4.7 WEB SERVER

TOMCAT SERVER

The Tomcat web server supports the JSP technology. This was the server used to develop the system. Tomcat is a very simple, but functional web server. It also does not occupy much disk space and it is very easy to start and stop the web server. Any web server, which supports the JSP technology, can also be used.

2.4.8 APPLICATION SERVER

WEBLOGIC SERVER

The Tomcat web server supports the JSP technology as well as EJB technology. This was the server used to develop the system. Weblogic is a very simple, but functional Application server. It also does not occupy much disk space and it is very easy to start and stop the web server. Any web server, which supports the JSP technology, can also be used.

2.4.9 STRUTS FRAMEWORK

The Struts framework provides the flexibility to develop the much less coupled applications. It generalizes and strictly implements MVC-model View Controller Architecture. That is the basic need of our architecture.

2.4.10 HIBERNATE

Hibernate is an object-relational mapping (ORM) library for the Java language, providing a framework for mapping an object-oriented domain model to a traditional relational database. Hibernate solves object-relational impedance mismatch problems by replacing direct persistence-related database accesses with high-level object handling functions.

Hibernate's primary feature is mapping from Java classes to database tables (and from Java data types to SQL data types). Hibernate also provides data query and retrieval facilities. Hibernate generates the SQL calls and attempts to relieve the developer from manual result set handling and object conversion and keep the application portable to all supported SQL databases with little performance overhead.

3. PROJECT PLANNING

Guidelines for Project Plans

- Use project plans to coordinate rather than to control.
- Make use of different personalities within the project environment.
- Prescheduled frequent revisions to project plans.
- Empower workers to estimate their own work.
- Describe value-creating tasks rather than activities.
- Define specific and tangible milestones.
- Use check lists, matrices, and other supplements to project plans

In the initial stage of project planning, the internal and external factors that influence the project should be determined and given priority weights. Examples of influences include the following:

❖ **Internal Factors**

- **Infrastructure:** Not an issue. One client machine and a server can execute our plan
- **Project scope:** Not satellite communication so restricted real time facility.
- **Project location:** Simple p4 systems and a server located anywhere in the world.
- **Project leadership:** governed by
- **Management approach:** Prototyping is used to manage project
- **Resource and capital availability:** Simple PCs and some legal software's for development.

❖ **External Factors**

- **Public needs:** very extensive
- **Market needs:** much extensive
- **National goals:** to increase public sharing in business

- **State of technology:** latest and upgraded
- **Industrial competitors:** much competition
- **Government regulations:** all regulation must be followed strictly.

4. DETAIL DESCRIPTION OF PROJECT

4.1 Existing System

Existing system is not CRM based and it cannot be generalized for other organizations. Modules are interdependent and we cannot use them independently. It was difficult to work with increasing bulk of data, which become difficult to handle and manage. It is designed by MIND Company only for the South City Ford organization and is not generalized for any other organization. In the existing system we have complex mechanism which uses dozens of spreadsheets and databases and other tools.

4.2 Proposed System

The basic architecture of the project is divided into three layers as described below:

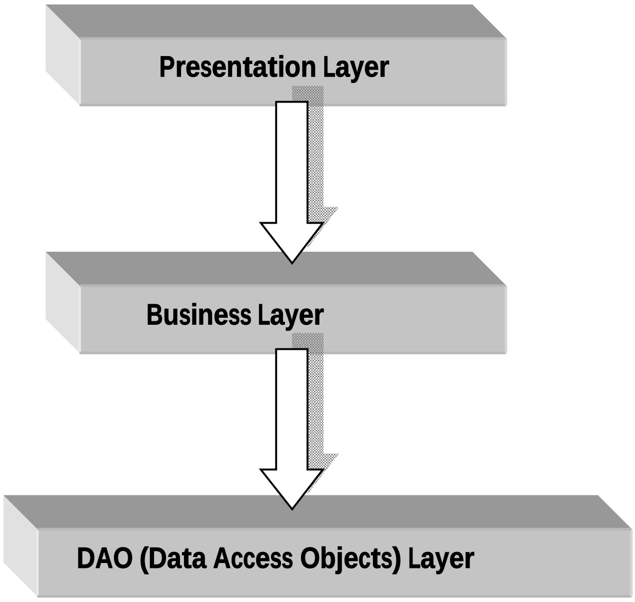


Fig. Layered description of Core ePortal-SRFQ

What is Core ePortal-SRFQ

This Core ePortal-SRFQ would be a CRM enables businesses of any size to manage every aspect of the business and customer lifecycle. Keep track of leads, prospects and sales is a huge challenge for any sales professional today. Track of assets and resources of any organization could be possible with the help of different customizable modules. Many will develop complex mechanisms using files and spreadsheets and databases and dozens of other tools. When it comes down to it, one of the best ways to keep your leads and sales organized effectively is to look for tracking software specifically designed for business organizations.

With well designed Core ePortal-SRFQ software, you can track leads and prospects and review and report on your sales. There are hundreds of options for this available, so doing a bit of research on your choices is a great first step. It is end to end comprehensive web enabled component based distributed Core ePortal-SRFQ to meet client's business challenge. Prime business objective is to provide a fast and secure platform to collect, manage improve and administer the sales data from the disparate distributor located all over India

Advantages of Core ePortal-SRFQ

- Better returns
- Huge Choice
- Familiarity
- Generalized

Disadvantages of Core ePortal-SRFQ

- Leverage
- Pattern Day Trader Rules
- Uptick Rule on Short Selling
- Costs

Customer relationship management

Customer relationship management is a broadly recognized, widely-implemented strategy for managing and nurturing a company's interactions with clients and sales prospects. It involves using technology to organize, automate, and synchronize business processes—principally sales activities, but also those for marketing, customer service, and technical support. The overall goals are to find, attract, and win new clients, nurture and retain those the company already has, entice former clients back into the fold, and reduce the costs of marketing and client service. Once simply a label for a category of software tools, today, it generally denotes a company-wide business strategy embracing all client-facing departments and even beyond. When an implementation is effective, people, processes, and technology work in synergy to increase profitability, and reduce operational costs.

CRM Benefits

These tools have been shown to help companies attain these objectives:

- Streamlined sales and marketing processes
- Higher sales productivity
- Added cross-selling and up-selling opportunities
- Improved service, loyalty, and retention
- Increased call center efficiency
- Higher close rates
- Better profiling and targeting
- Reduced expenses
- Increased market share
- Higher overall profitability
- Marginal costing

Technologies and Architecture used:

Software Life cycle Model: Prototype Model

Platform: Microsoft Windows XP and its latest versions

Primary Interface/Frame (Front End): HTML, XHTML, XML, Ajax, JavaScript

Core Technology: Core Java and J2EE

Core Framework: Struts 1.2.9

Environment: Net Beans IDE 6.7.1, Dreamweaver 8.0

Web Server: Apache Tomcat 6.0

Application Server: BEA Web Logic Application Server

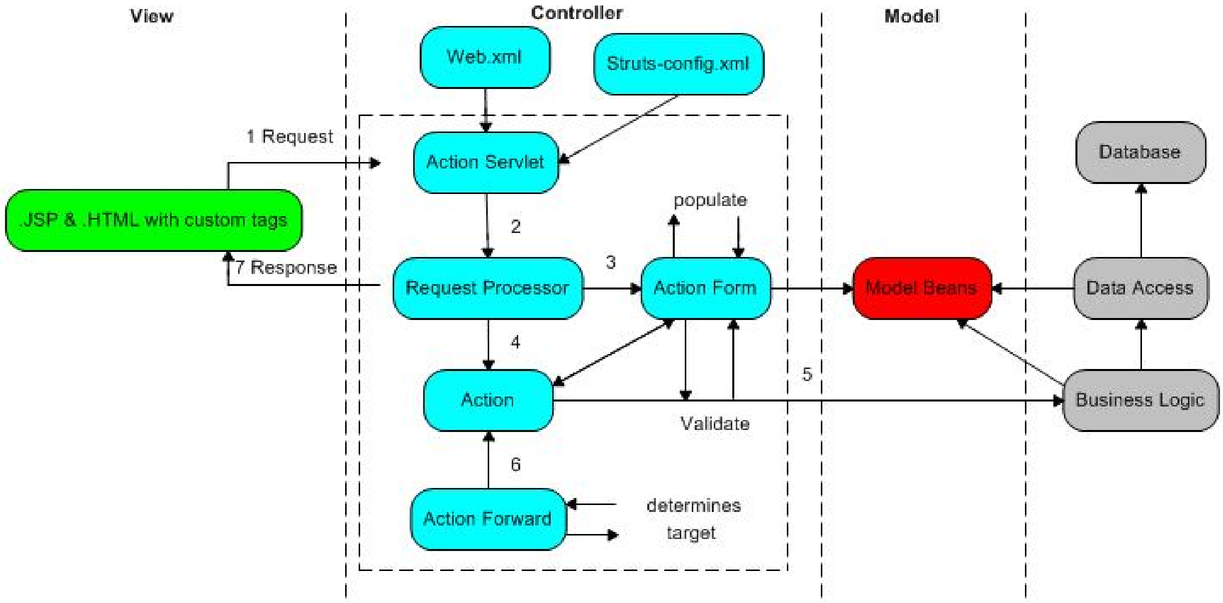
Connectivity: Hibernate, JDBC (Java database connectivity)

Secondary Interface (Back End): MySql 5.0

Application Components: Java Beans and Enterprise Java Beans

Core Architecture: MVC (Model-view-controller)

Product Perspective:



This Application is developed on Struts Framework and Sturts Framework follow MVC 2 Architecture.

The web pages (XHTML/JSP) are present to provide the user interface on client side. Communication between client and server is provided through HTTP/HTTPS protocols.

4.3 Modules in Project

1. Configuration(Admin)
2. Security(Admin)
3. Request Specification(Production)
4. Configure Supplier(Admin)
5. Quotation Posting(Supplier)
6. Technical Analysis(Technical)
7. Financial Analysis(Financial)
8. Admin Approval(Admin)
9. Sample Request(Production)
10. Supplier Sampling Confirmation & Evaluation(Supplier)
11. Sample Technical Approval(Technical)
12. Sample Financial Approval(Financial)
13. Sample Operational Approval(Production)
14. Admin Sample Approval(Admin)

5. SYSTEM DESIGN

This project has to design and implement a web application for the purpose of Security. The system will be available to everyone via Internet. The system must be designed to retrieve data from a database server and display the results in a formatted manner on any browser. Any modifications to the data must also be done through an interface provided by the browser.

System design provides the understanding and procedural details necessary for implementing the system recommended in the system study .Emphasis is on translating the performance requirements into design specifications . The Design phase is a transition from a user – oriented document (System proposal) to a documented oriented to the programmers or database personnel.

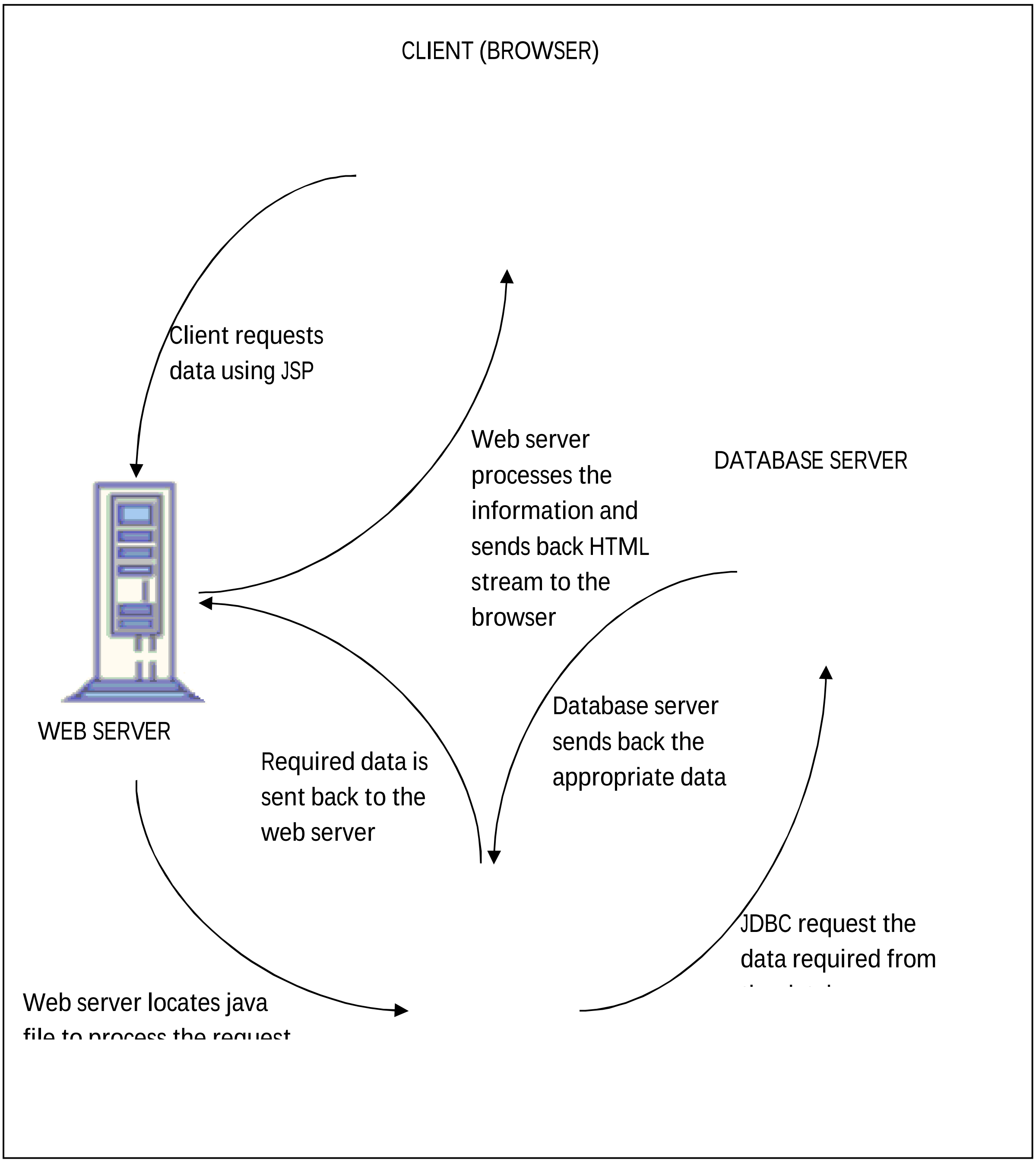
System design goes through following phases of development:-

- Physical design
- Logical design
- Database & file design
- Interface design

5.1 Functional Architecture Design / Physical design

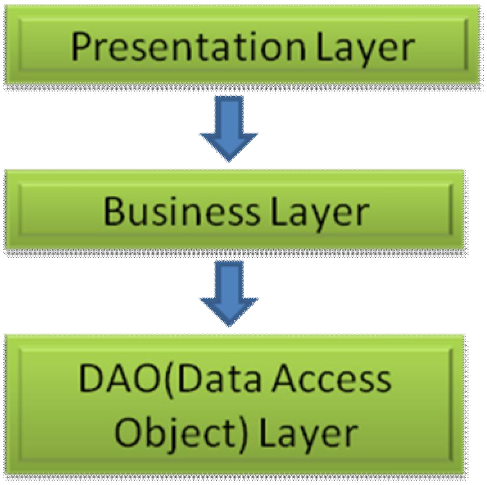
The system will be designed using the three tier model. The database server will form one tier of the three tiers. The database server used will be Mysql 5.0. The next tier is the component which will retrieve data from the database. The component used for this purpose will be Java language with its rich data communicating abilities. The JDBC – ODBC Bridge will be used for interlinking Mysql 5.0 and Java. The last tier is the Web Server which will process the requests from the user front end, i.e. the browser and translate them into calls to the Java language. The web server used for this project is the JSP server. For purposes of developing this project Tomcat Server will be used, though any JSP server can be used as the web server.

The web server will form the interface between the JSP scripting and Java functions. This tier will acts as a proxy for all client requests. In this environment, the client requests are translated into calls to the java functions. These java functions will communicate with the database server and perform the required data handling and then return data back to the web server. These data are then sent back to the client by the web server after converting them into HTML statements, which are displayed by the internet browser. The way this arrangement works can be explained using the figure


CLIENT (BROWSER)
Client requests data using JSP
Web server processes the information and sends back HTML stream to the browser
DATABASE SERVER
WEB SERVER
Database server sends back the appropriate data
Required data is sent back to the web server
JDBC request the data required from
Web server locates java file to process the request

Architectural Design

The basic architecture for the proposed system is like that

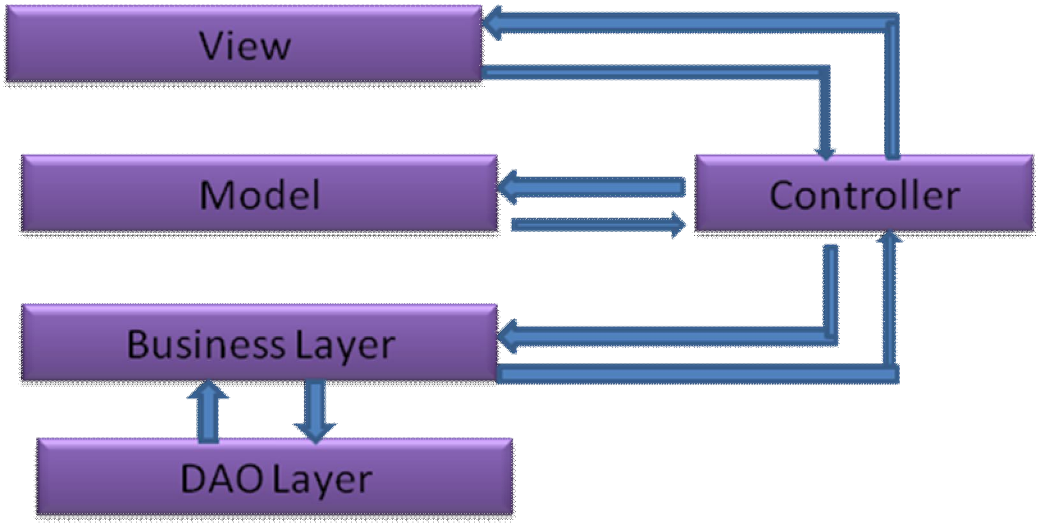


Presentation Layer: All the Jsp and html pages are designed in this tier

Business Layer: All the business logic and service code are written here.

DAO Layer: All the database connections and result set related codes are written here

Detailed Design uses standard MVC Architecture



The Model-View-Controller Design Pattern

A *design pattern* is a series of objects and object relationships that provide a proven, extensible solution to a particular software design problem. The Model-View-Controller (MVC) pattern is arguably the best known, most famous design pattern of them all.

MVC was originally developed in the late 1970s at the Xerox Palo Alto Research Center (PARC). It was originally built to manage the GUI and user interaction on some of the first window-based computers (another innovation from the PARC—in addition to Ethernet, local area networks, mice for input devices, and numerous other firsts).

The design problem that MVC solves is that of simplifying three primary functions that are common in many applications:

- Maintaining the data in a back-end store or remote system
- Building the end-user presentation layer
- Maintaining the conditional logic that decides which screens are presented to the user, what happens when errors occur, and exactly how and when the remote systems are updated

It is possible to combine all this processing into a single module and get a system to work. (In fact, a significant amount of early JSP development did exactly that!) Problems primarily occur when you try to perform maintenance on the code. In the case of JSP, this is compounded by the fact that the HTML designers who maintain the look and feel of the application are different people (and have different skill sets) from those who maintain the Java code that controls the processing.

MVC addresses this problem by separating the code into three distinct areas:

- **Model** components that maintain data in a back-end store or remote system
- **Views** that build the end-user presentation layer
- **Controllers** to maintain conditional logic that decides which screens are presented to the user, what happens when errors occur, and exactly how and when the remote systems are updated

MVC simplifies maintenance by keeping all this logic from becoming intertwined. It allows the details of each piece to be hidden from the others and reduces the coding linkages between them. This is how MVC provides a natural boundary between the people who write the Java and the people who maintain the HTML and presentation layer.

A good example of this is in how MVC can simplify exception processing. Imagine that after a user logs in, you send a request to a remote system to fetch the user's customer information. What do you do if the remote system is unavailable? In normal JSP processing, it's common to embed logic at the top of your JSP file to detect this and change what you display to the user when the problem occurs. Using MVC, you can pull this logic out of the JSP page altogether: You create a page dedicated to presenting the error message and have the Controller determine which page to send the user to. If the remote system is available, the user gets the first page. If not, the Controller sends him to the error page.

This approach to exception processing has multiple benefits. The first comes from the fact that, on many pages, multiple types of exceptions must be handled. Having a single JSP page that detects all possible errors and presents a different message when each error happens can become complicated fast. Moving that logic into a Controller makes things easier to maintain: The logic is maintained in the Controller, and only the presentation is maintained in the JSP file.

Of course, another primary benefit of pulling the exception logic out of the main JSP pages is that it makes maintaining the JSP pages easier!

These benefits really extend to all forms of conditional processing. Here are some other examples:

If different Views are required depending on what data is retrieved from a database or remote system (for example, products on sale versus products not on sale), the Controller component can make the decision about which page to present. This keeps the logic out of the JSP page itself.

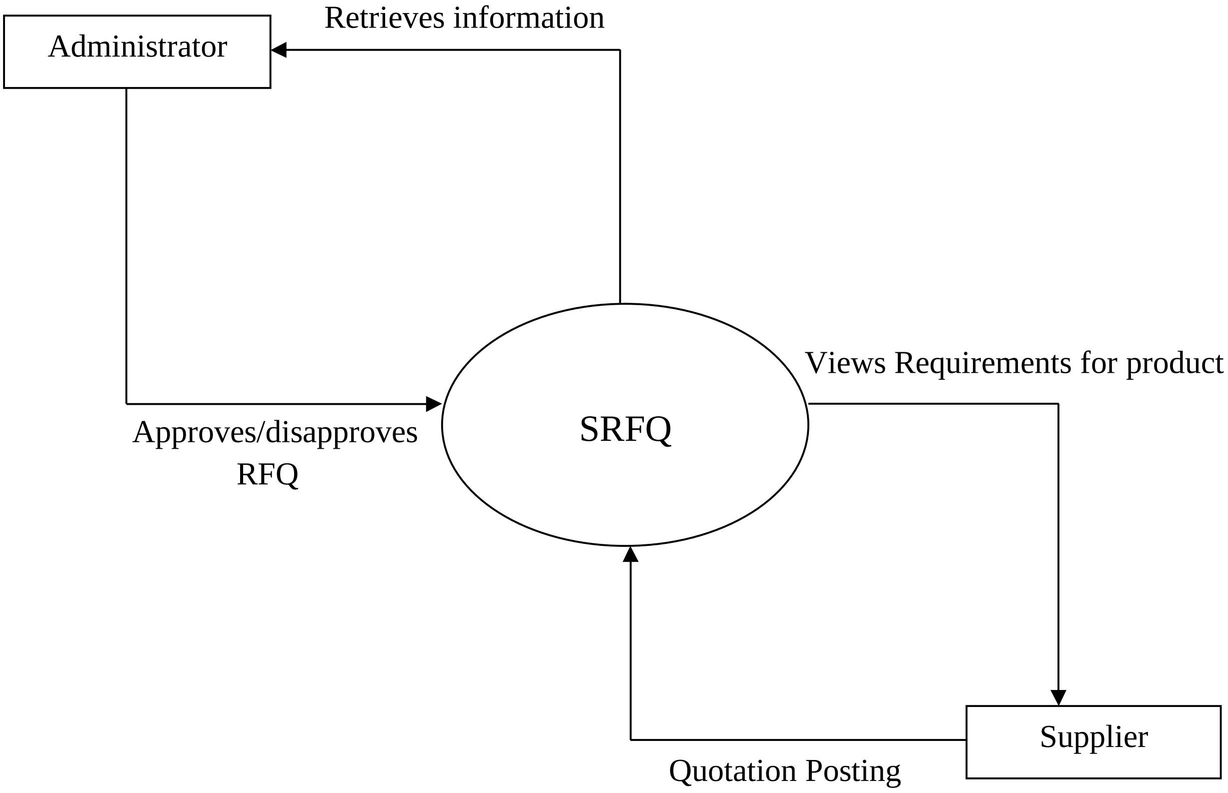
If your site changes based on either the time of day or the day of the week, that logic is easy to implement in the Controller. You simply have the Controller check the date and forward the user to the appropriate page.

Sometimes a data entry process can span several pages, some of which are optional. An example of this is signing up for insurance: You need to be shown the data entry pages for dependents only if you choose family coverage. In cases like this, MVC makes it easy to control the flow of pages that are shown to the user. Trying to embed this logic into the JSP pages makes things much more complex.

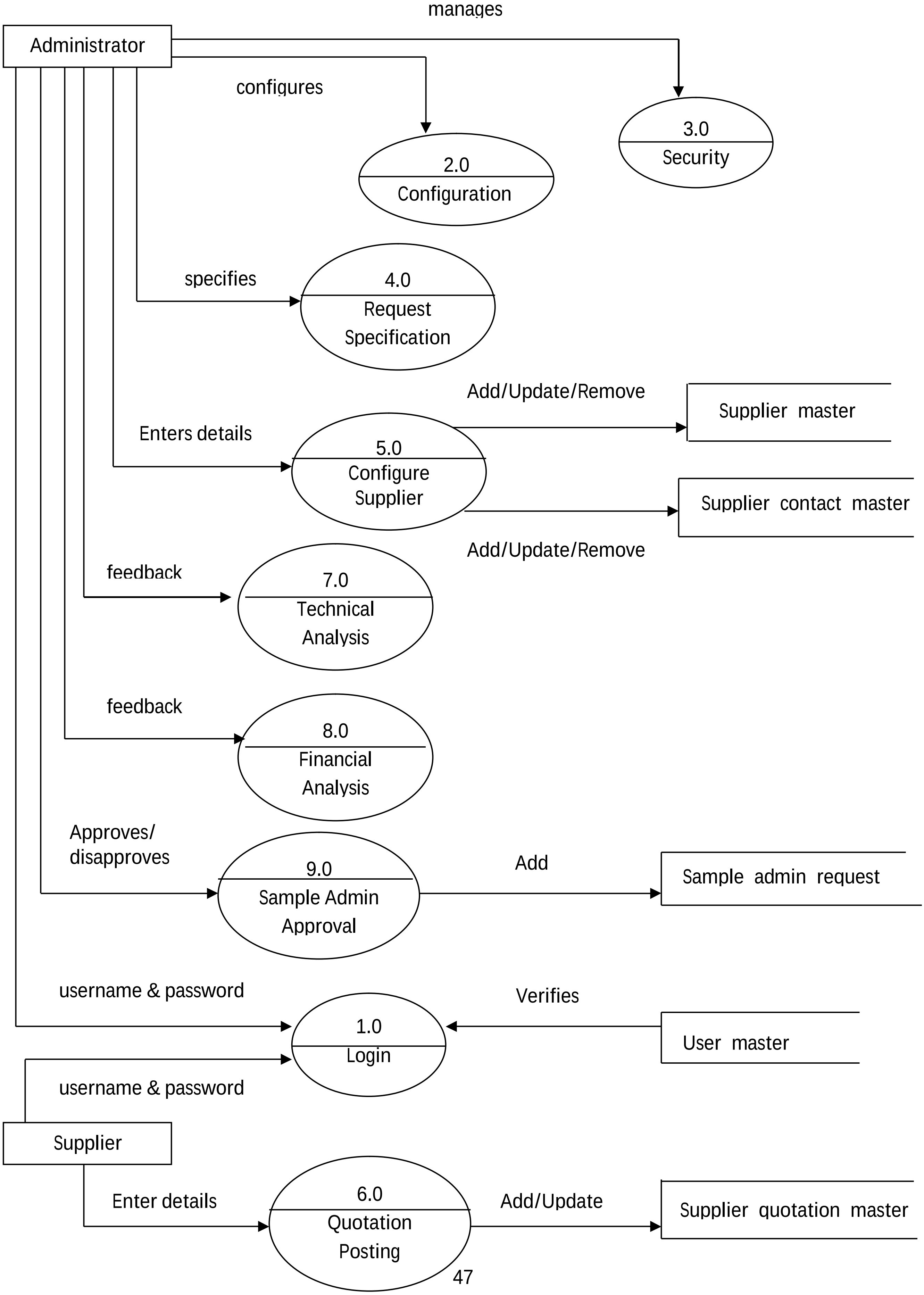
5.2 Logical Design

5.2.1 Data Flow Diagram

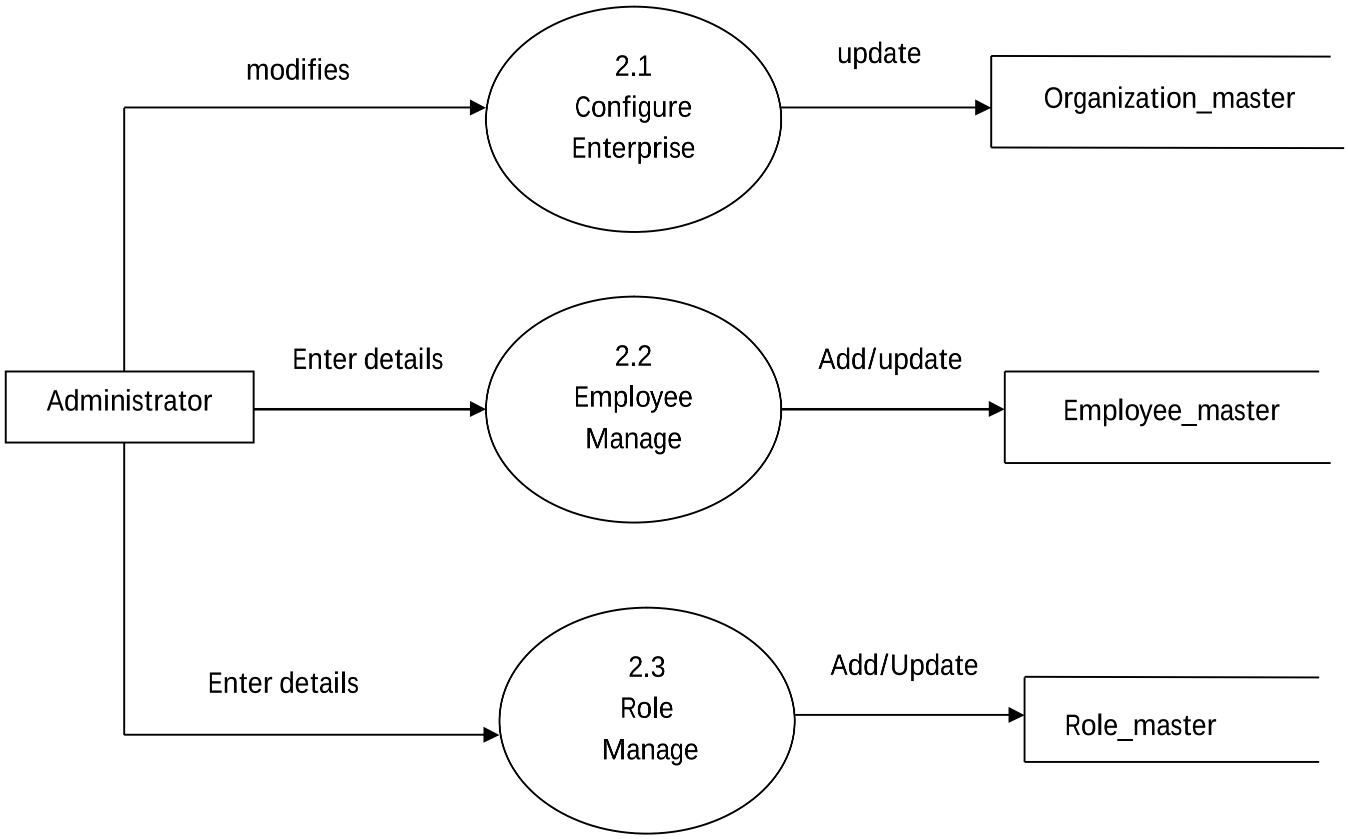
CONTEXT LEVEL DFD



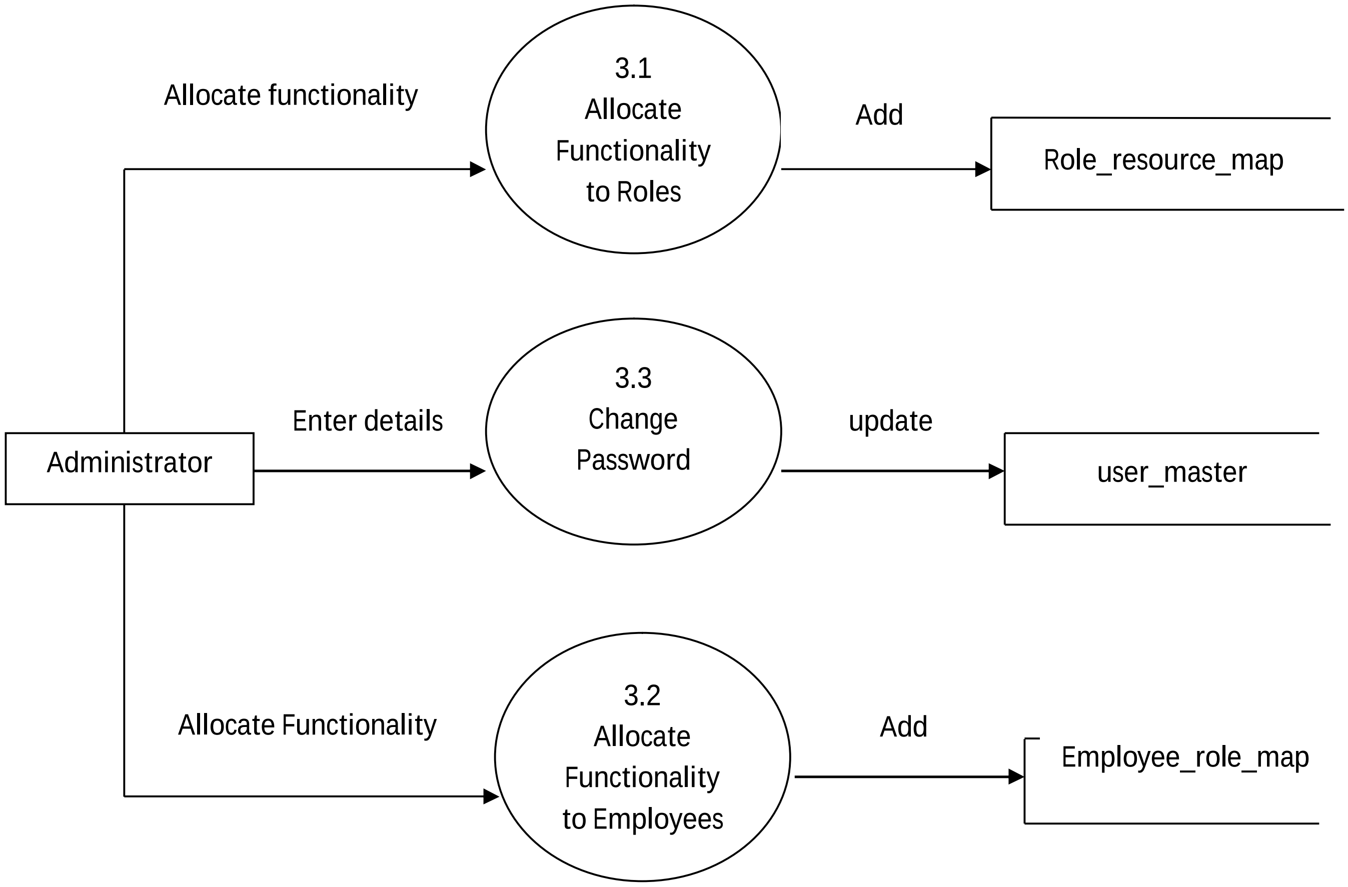
LEVEL 1 DFD


manages
Administrator
configures
3.0
Security
2.0
Configuration
specifies
4.0
Request Specification
Add/Update/Remove
Supplier master
Enters details
5.0
Configure Supplier
Supplier contact master
Add/Update/Remove
feedback
7.0
Technical Analysis
feedback
8.0
Financial Analysis
Approves/ disapproves
9.0
Sample Admin Approval
Add
Sample admin request
username & password
Verifies
1.0
Login
User master
username & password
Supplier
Enter details
6.0
Quotation Posting
Add/Update
Supplier quotation master

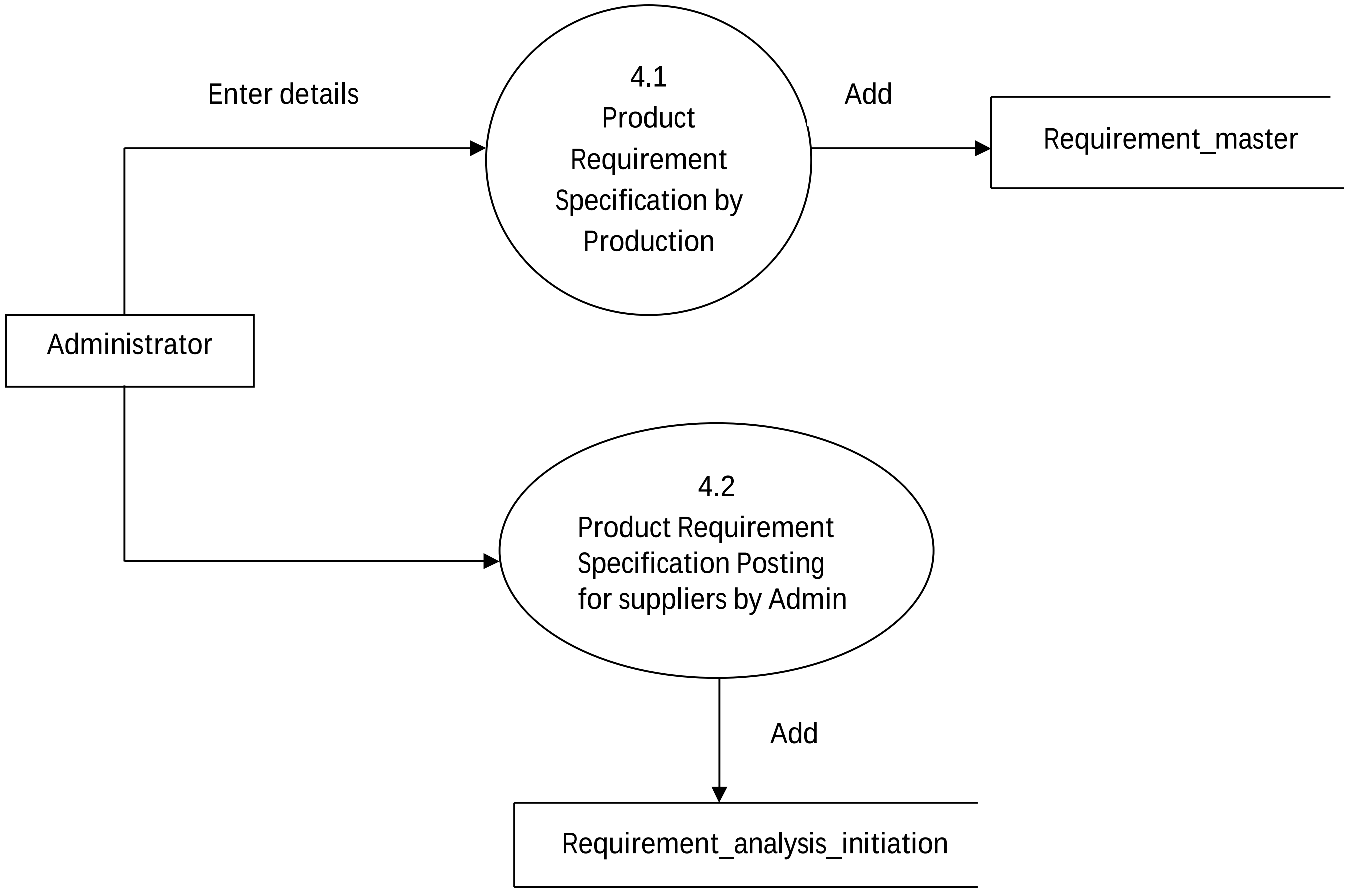
LEVEL 2 DFD FOR CONFIGURATION



LEVEL 2 DFD FOR SECURITY



LEVEL 2 DFD FOR REQUIREMENT SPECIFICATION



5.2.2 Entity Relationship Diagram

In order to make the physical design, the data is first analyzed through ER (Entity Relationship) modeling technique. This technique emphasizes that a database system for an application is made of entities. An entity is any comprehensive real world object that can be identified. According to its entities have relationship with each other an entity relationship diagram can represent various entities of any system.

5.3 Database Design

We have used relational data model as the model for our database design. Database is optimized up to Third Normal Form

The database has been implemented by using MySql Environment .This environment facilitates all the necessities for the database design as well as it take care of the integrity constraints and normalization phenomenon.

Normalization

Normalization provides for table optimisation through the investigation of entity relationships. Main purpose of normalization is to avoid Data redundancy and some unforeseen scalability factors. Normalization is done to remove Insertion, Updating and Modification anomalies and redundancy of data. A certain level of normalization of tables in database gives a particular normal form based of particulars steps followed. Database can be normalized up to any defined normal forms according as the need of application and its effectiveness.

Database is normalized up to **Third Normal Form.** Further normalization of database was not considered taking into account the need of application and ease of working with database.

The database is in **First Normal Form** as all the fields of all tables are atomic. There is no multivalued field in any table. Table Inquiry needs to store more than one inquiry officers but as it was known that not more than two inquiry officers are appointed, so two different fields for each inquiry officers are used. Hence the tables satisfy the constraint of atomicity of first normal form.

The database is in **Second Normal Form** as it satisfies the constraint of full functional dependency. All the fields of all tables are fully functional dependent on the primary key.

The database is in **Third Normal Form** as all its tables satisfy the constraint that there should be no transitive dependency. No field has transitive dependency on the key field. Thus database also satisfies the constraints of third normal form.

5.3.1 Description of Tables Used

- **Currency_master**

Column Name	Datatype	Details
Currency_id	VARCHAR(10)	Primary Key, Not Null
Currency_desc	VARCHAR(20)	Not Null

- **Employee_master**

Column Name	Datatype	Details
Emp_id	VARCHAR(10)	Primary Key, Not Null
Emp_Fname	VARCHAR(45)	Not Null
Emp_Lname	VARCHAR(45)	Not Null
Current_Street_Address	VARCHAR(45)	Not Null
Current_City	VARCHAR(45)	Not Null
Current_State	VARCHAR(45)	Not Null
Current_Zipcode	VARCHAR(10)	
Current_Country	VARCHAR(45)	Not Null
Current_Phone1	VARCHAR(13)	
Current_Phone2	VARCHAR(13)	Not Null
Current_Mob	VARCHAR(11)	Not Null
Current_Fax	VARCHAR(11)	Not Null
Perm_Street_Address	VARCHAR(45)	Not Null
Perm_City	VARCHAR(45)	Not Null
Perm_State	VARCHAR(45)	Not Null
Perm_Zipcode	VARCHAR(10)	
Perm_Country	VARCHAR(45)	Not Null
Perm_Phone1	VARCHAR(13)	
Perm_Phone2	VARCHAR(13)	Not Null
Perm_Mob	VARCHAR(11)	Not Null
Perm_Fax	VARCHAR(11)	Not Null
Email_id	VARCHAR(45)	Not Null
Date_Of_Joining	DATE	Not Null
Date_Of_leaving	DATE	
Status_Id	VARCHAR(10)	Foreign Key, Not Null

- **Employee_role_map**

Column Name	Datatype	Details
Emp_id	VARCHAR(10)	Foreign Key, Not Null
Role_id	VARCHAR(10)	Foreign Key, Not Null

- **Financial_feedback_master**

Column Name	Datatype	Details
Finance_feedback_id	VARCHAR(10)	Primary Key, Not Null
Supp_Quotation_id	VARCHAR(10)	Foreign Key, Not Null
Finance_Goodwill_id	VARCHAR(10)	Foreign Key, Not Null
Rating_id	VARCHAR(10)	Foreign Key, Not Null
Finance_Approved	ENUM(‘Yes’,’No’)	Not Null
Feedback	VARCHAR(45)	Not Null
Feedback_by	VARCHAR(45)	Not Null
Feedback_date	DATE	Not Null

- **Financial_goodwill_master**

Column Name	Datatype	Details
Finance_Goodwill_id	VARCHAR(10)	Primary Key, Not Null
Finance_Goodwill_desc	ENUM('Very Expensive', 'Expensive', 'Reasonable', 'Economical', 'NewEntry')	Not Null

- **Material_group_master**

Column Name	Datatype	Details
Material_Group_id	VARCHAR(10)	Primary Key, Not Null
Material_Group_Name	VARCHAR(45)	Not Null

- **Material_master**

Column Name	Datatype	Details
Material_Id	VARCHAR(10)	Primary Key, Not Null
Material_Desc	VARCHAR(45)	Not Null
Tech_Desc	VARCHAR(45)	Not Null
Material_Group_id	VARCHAR(10)	Foreign Key, Not Null

- **Material_tech_goodwill_master**

Column Name	Datatype	Details
Material_Goodwill_id	VARCHAR(10)	Primary Key, Not Null
Material_Goodwill_desc	ENUM('Excellent', 'Very Good', 'Good', 'Average',	Not Null

	'Low', 'Very Low', 'New Entry')	

- **Employee_role_map**

Column Name	Datatype	Details
Org_Id	VARCHAR(10)	Primary Key, Not Null
Org_Name	VARCHAR(60)	Not Null
Org_Add	VARCHAR(60)	Not Null
State	VARCHAR(30)	Not Null
Zipcode	VARCHAR(10)	Not Null
Country	VARCHAR(30)	Not Null
Phone1	VARCHAR(13)	Not Null
Phone2	VARCHAR(13)	Not Null
Fax	VARCHAR(13)	Not Null
Email	VARCHAR(45)	Not Null

- **Rating_master**

Column Name	Datatype	Details
Rating_id	VARCHAR(10)	Primary Key, Not Null
Rating_desc	ENUM('Very High', 'High', 'Average', 'Low', 'Very Low')	Not Null

- **Requirement_analysis_initiation**

Column Name	Datatype	Details
Req_id	VARCHAR(10)	Primary Key, Not Null
Status_id	VARCHAR(10)	Foreign Key, Not Null
Status_changed_by	VARCHAR(30)	Not Null
Status_changed_date	DATE	Not Null
Remark	VARCHAR(45)	Not Null

- **Requirement_master**

Column Name	Datatype	Details
Req_id	VARCHAR(10)	Primary Key, Not Null
Material_Id	VARCHAR(10)	Foreign Key, Not Null
Quantity	VARCHAR(20)	Not Null
Unit_Id	VARCHAR(10)	Foreign Key, Not Null

Req_posting_Opening_Date	DATE	Not Null
Req_posting_Closing_Date	DATE	Not Null
Posted_By	VARCHAR(45)	Not Null
R_Priority_Id	VARCHAR(10)	Foreign Key, Not Null

- **Requirement_priority_master**

Column Name	Datatype	Details
R_priority_id	VARCHAR(10)	Primary key, Not Null
R_priority_desc	ENUM('Normal', 'Average', 'High', 'Crucial')	Not Null

- **Resource_master**

Column Name	Datatype	Details
Resc_id	VARCHAR(10)	Primary key, Not Null
Group_English	VARCHAR(45)	Not Null
Resc_Name_English	VARCHAR(45)	Not Null
Link	VARCHAR(45)	Not Null
Show_in_menu	VARCHAR(45)	Not Null

- **Role_master**

Column Name	Datatype	Details
Role_id	VARCHAR(10)	Primary key, Not Null
Role_Name	VARCHAR(45)	Not Null
Role_Desc	VARCHAR(45)	Not Null
Org_id	VARCHAR(10)	Foreign Key, Not Null
Status_id	VARCHAR(10)	Foreign Key, Not Null

- **Role_resource_map**

Column Name	Datatype	Details
Role_Resource_Map_id	VARCHAR(10)	Primary key, Not Null
Role_id	VARCHAR(10)	Foreign Key, Not Null
Resc_id	VARCHAR(10)	Foreign Key, Not Null

- **Salutation_master**

Column Name	Datatype	Details
Salutation_id	VARCHAR(10)	Primary Key, Not Null
Salutation_Desc	ENUM('Mr','Mrs','Miss')	Not Null

- **Sample_admin_request**

Column Name	Datatype	Details
Sample_id	VARCHAR(10)	Primary Key, Not Null
Supp_Quotation_id	VARCHAR(10)	Foreign Key, Not Null

- **Status_master**

Column Name	Datatype	Details
Status_id	VARCHAR(10)	Primary key, Not Null
Status_Desc	ENUM('Active','InActive')	Not Null

- **Supplier_contact_master**

Column Name	Datatype	Details
Contact_person_id	VARCHAR(10)	Primary key, Not Null
Supp_id	VARCHAR(10)	Foreign Key, Not Null
Salutation_id	VARCHAR(10)	Foreign Key, Not Null
Contact_person_fname	VARCHAR(30)	Not Null
Contact_person_lname	VARCHAR(30)	Not Null
Contact_person_function	VARCHAR(30)	
Contact_person_phone	VARCHAR(13)	Not Null
Contact_person_email	VARCHAR(30)	Not Null
Termsandconditions	VARCHAR(60)	

- **Supplier_master**

Column Name	Datatype	Details
Supp_id	VARCHAR(10)	Primary key, Not Null
Supp_name	VARCHAR(45)	Not Null
Supp_status_id	VARCHAR(10)	Foreign Key, Not Null

- **Supplier_quotation_master**

Column Name	Datatype	Details
Supp_Quotation_id	VARCHAR(10)	Primary key, Not Null
Supp_id	VARCHAR(10)	Foreign Key, Not Null
Req_id	VARCHAR(10)	Foreign Key, Not Null
Proposedcost	VARCHAR(10)	Not Null
Currency_id	VARCHAR(10)	Foreign Key, Not Null
Unit_id	VARCHAR(10)	Foreign Key, Not Null
Tech_Specification	VARCHAR(45)	Not Null
Remarks	VARCHAR(45)	Not Null
Tech_Feedback_Status	ENUM('Pending','Done')	Not Null
Finance_Feedback_Status	ENUM('Pending','Done')	Not Null

- **Supplier_status_master**

Column Name	Datatype	Details
Supp_status_id	VARCHAR(10)	Primary Key, Not Null
Supp_status_desc	ENUM('External','Internal')	Not Null

- **Technical_feedback_master**

Column Name	Datatype	Details
Tech_feedback_id	VARCHAR(10)	Primary Key, Not Null
Supp_Quotation_id	VARCHAR(10)	Foreign Key, Not Null
Material_Goodwill_id	VARCHAR(10)	Foreign Key, Not Null
Rating_id	VARCHAR(10)	Foreign Key, Not Null
Tech_Approved	ENUM(‘Yes’,’No’)	Not Null
Feedback	VARCHAR(45)	Not Null
Feedback_by	VARCHAR(45)	Not Null
Feedback_date	DATE	Not Null

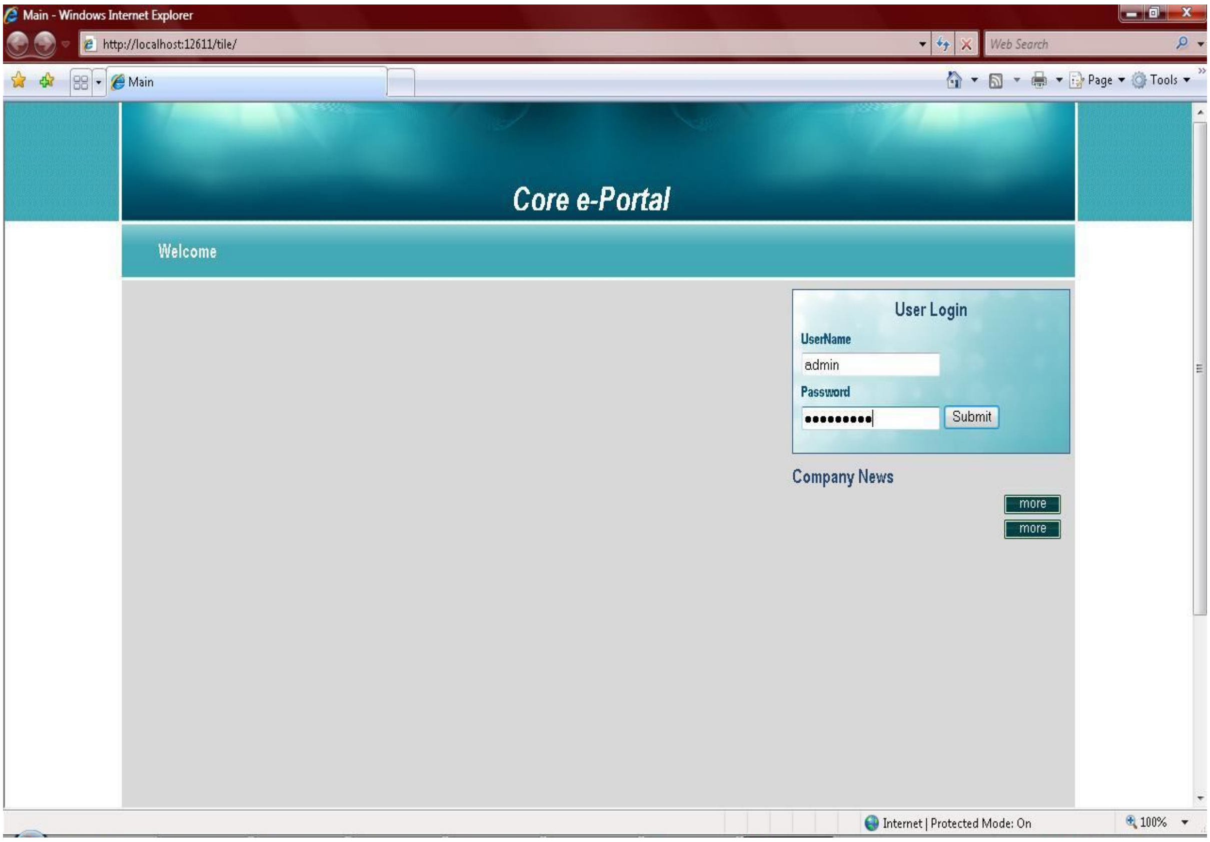
- **Unit_measure_master**

Column Name	Datatype	Details
Unit_id	VARCHAR(10)	Primary Key, Not Null
Unit_of_Measure	VARCHAR(20)	Not Null

- **User_master**

Column Name	Datatype	Details
User_id	VARCHAR(10)	Primary Key, Not Null
Emp_id	VARCHAR(20)	Foreign Key, Not Null
Login_name	VARCHAR(30)	Not Null
Password	VARCHAR(30)	Not Null

6. USER INTERFACE SNAPSHOTS

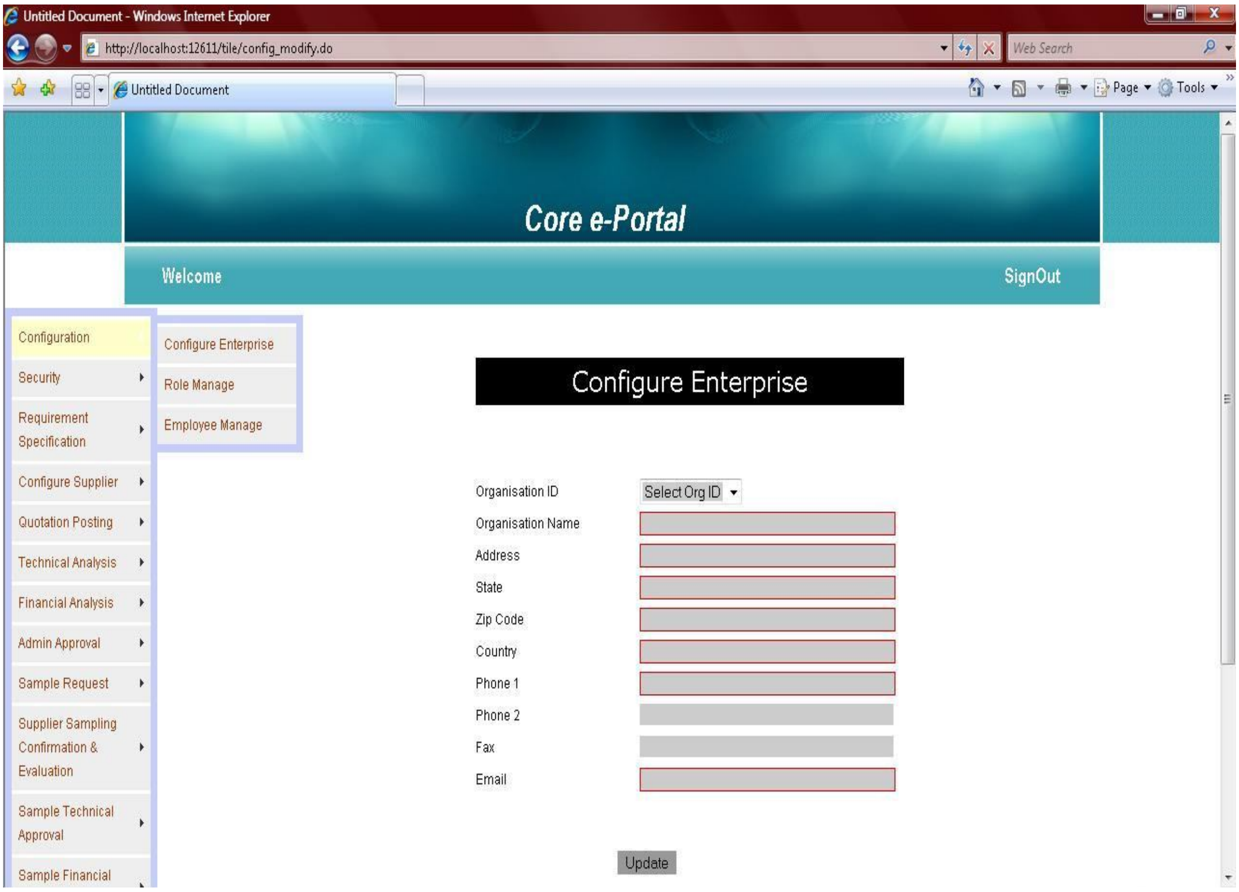

Main - Windows Internet Explorer
http://localhost:12611/tile/
Web Search
Main
Page
Tools
Core e-Portal
Welcome
User Login
Submit
Company News
more
more
Internet | Protected Mode: On
100%

UserName: admin

Password: •••••••••

Login Page:

The user enters the username and password to view further portal.

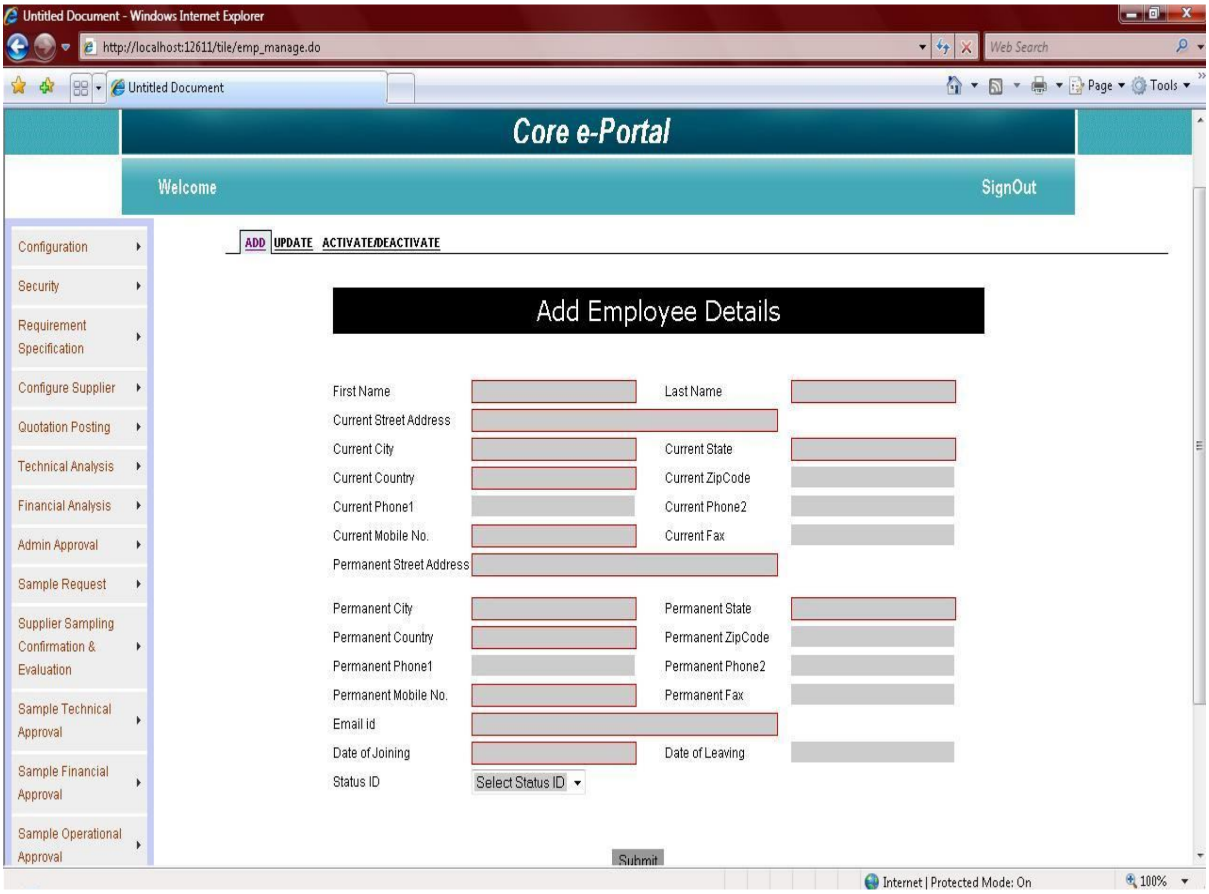


Configure Enterprise

Organisation ID: Select Org ID
Organisation Name
Address
State
Zip Code
Country
Phone 1
Phone 2
Fax
Email

Configure Enterprise:

The admin has the privilege of modifying the details of the enterprises linked with the organization. When the organization ID is selected from the drop down list all the details of the enterprise are displayed and any changes can be made and saved.

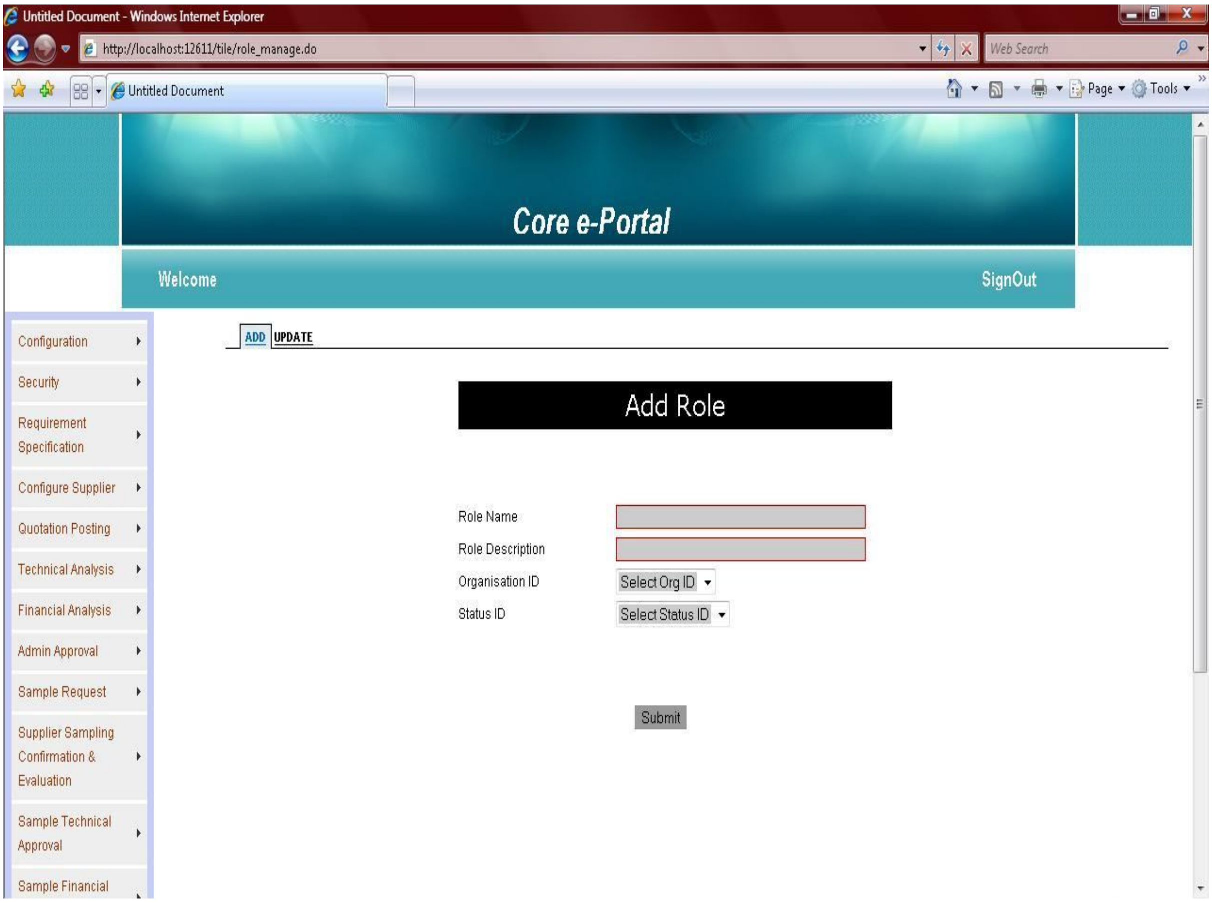

Untitled Document - Windows Internet Explorer
http://localhost:12611/tile/emp_manage.do
Web Search
Untitled Document
Page
Tools
Core e-Portal
Welcome
SignOut
Configuration
Security
Requirement Specification
Configure Supplier
Quotation Posting
Technical Analysis
Financial Analysis
Admin Approval
Sample Request
Supplier Sampling Confirmation & Evaluation
Sample Technical Approval
Sample Financial Approval
Sample Operational Approval
ADD
UPDATE
ACTIVATE/DEACTIVATE
Submit
Internet | Protected Mode: On
100%

Add Employee Details

First Name
Last Name
Current Street Address
Current City
Current State
Current Country
Current ZipCode
Current Phone1
Current Phone2
Current Mobile No.
Current Fax
Permanent Street Address
Permanent City
Permanent State
Permanent Country
Permanent ZipCode
Permanent Phone1
Permanent Phone2
Permanent Mobile No.
Permanent Fax
Email id
Date of Joining
Date of Leaving
Status ID: Select Status ID

Employee Manage

The admin has the option of adding, deleting and activating/deactivating employee status. All the details entered are stored in the database in table empdetails.



Add Role

Role Name

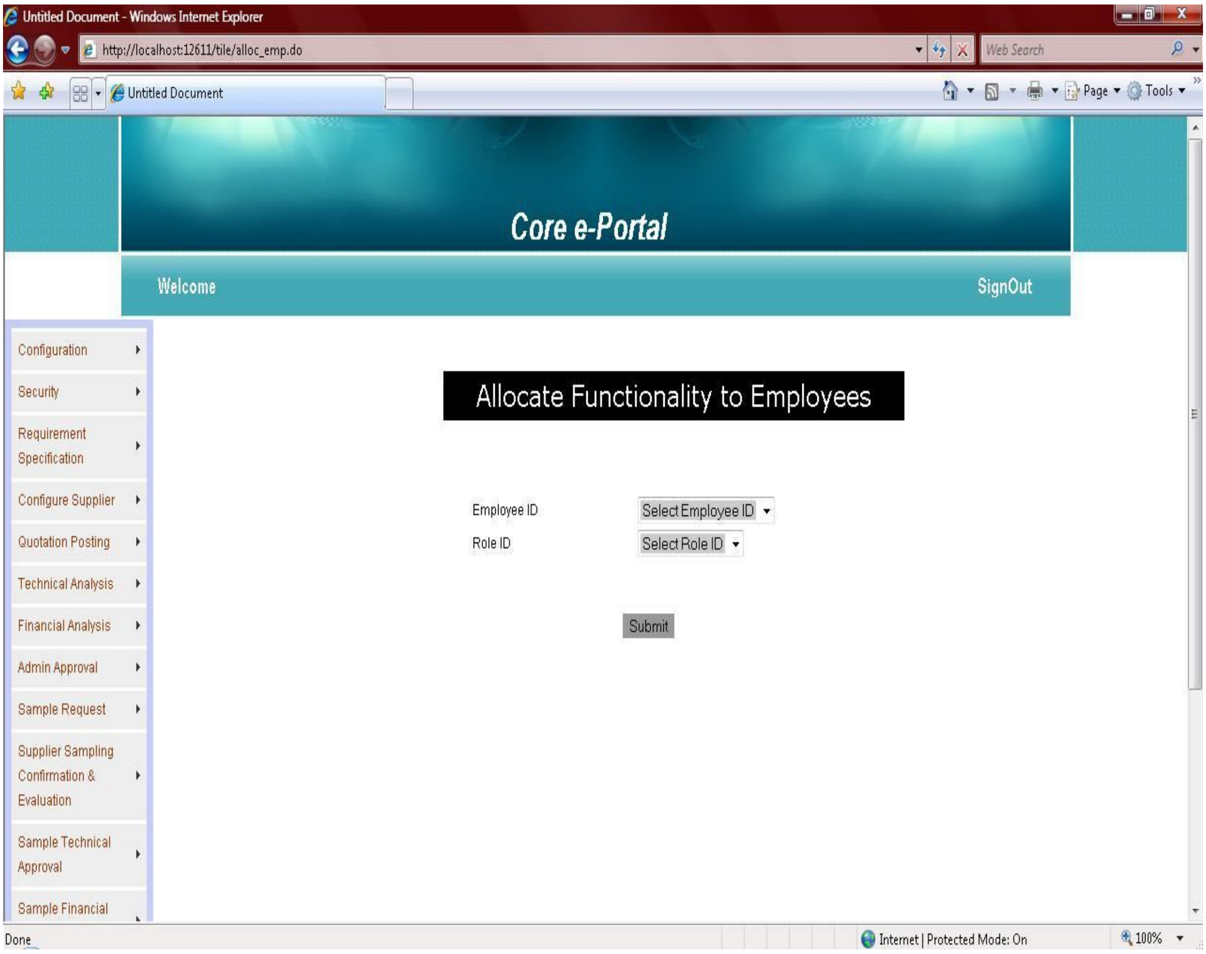
Role Description

Organisation ID: Select Org ID

Status ID: Select Status ID

Role Manage

The Admin can add different roles in the organization from this page. Admin can select which department has to be allotted which role.

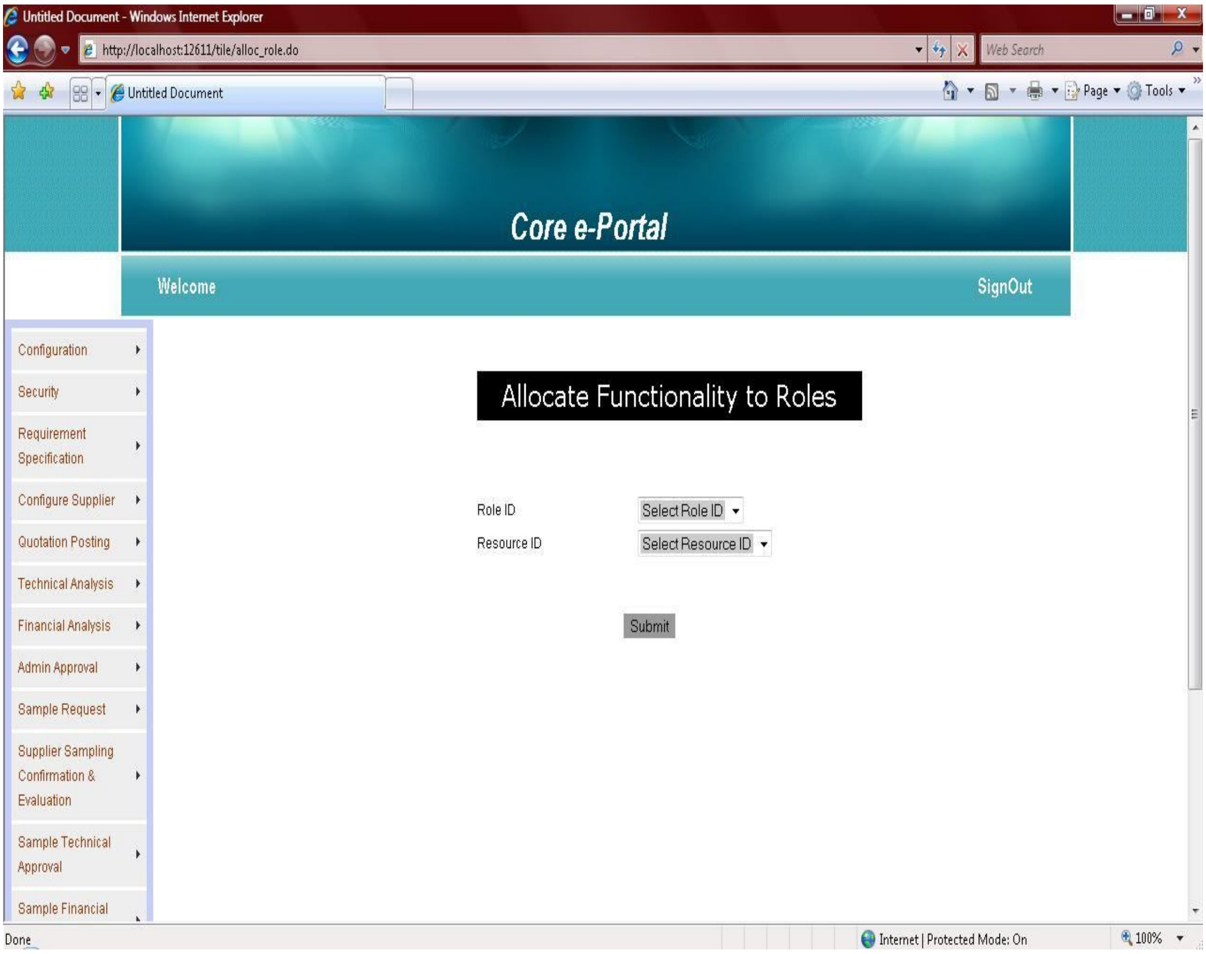


Allocate Functionality to Employees

Employee ID: Select Employee ID

Role ID: Select Role ID

The roles allocated to all the employees are entered through this part of the project.

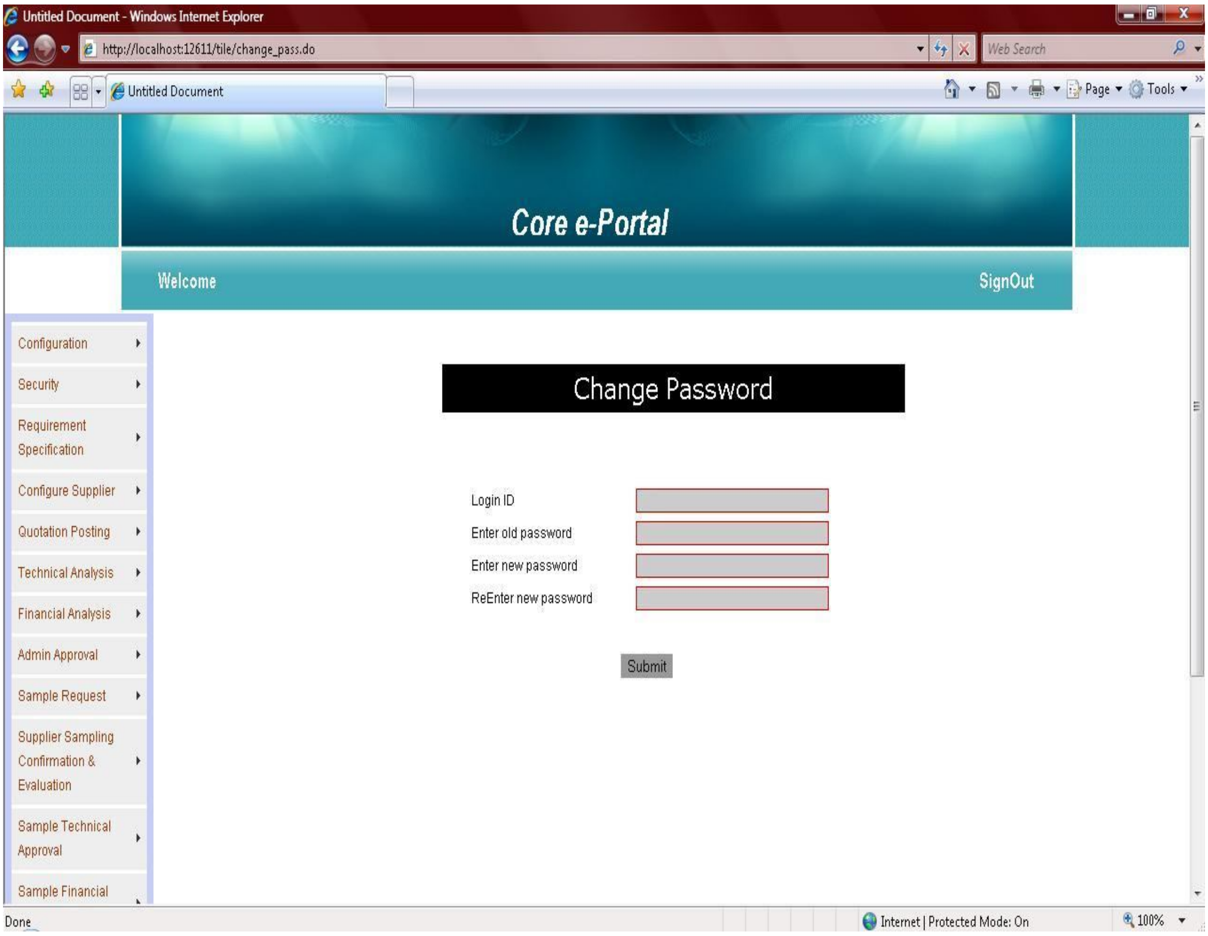


Allocate Functionality to Roles

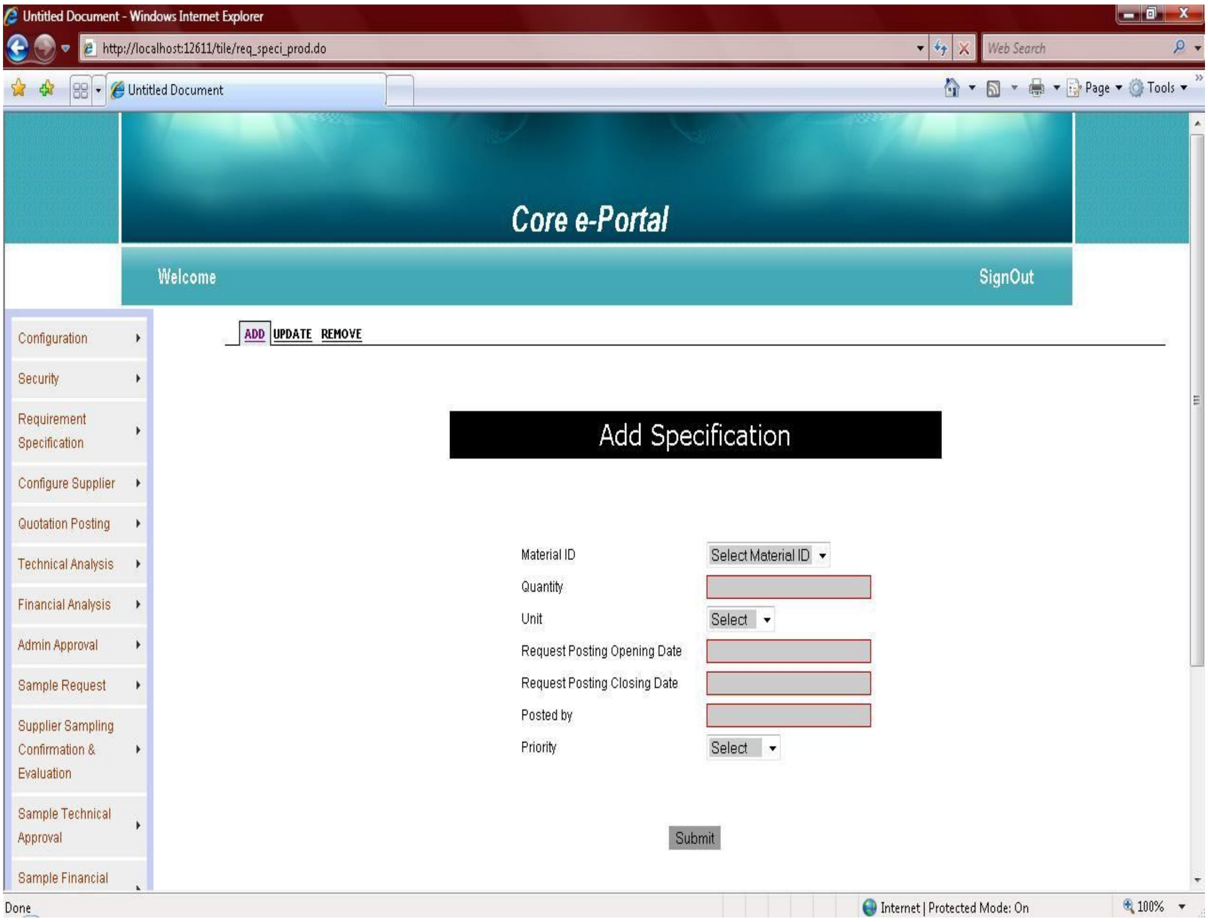
Role ID: Select Role ID

Resource ID: Select Resource ID

Various roles are allocated functionalities from the drop down list.



This is the security part of the project. If the admin wants to change the password of his account he can enter the old password and choose a new password. If the old password matches his password is set to the new password entered.

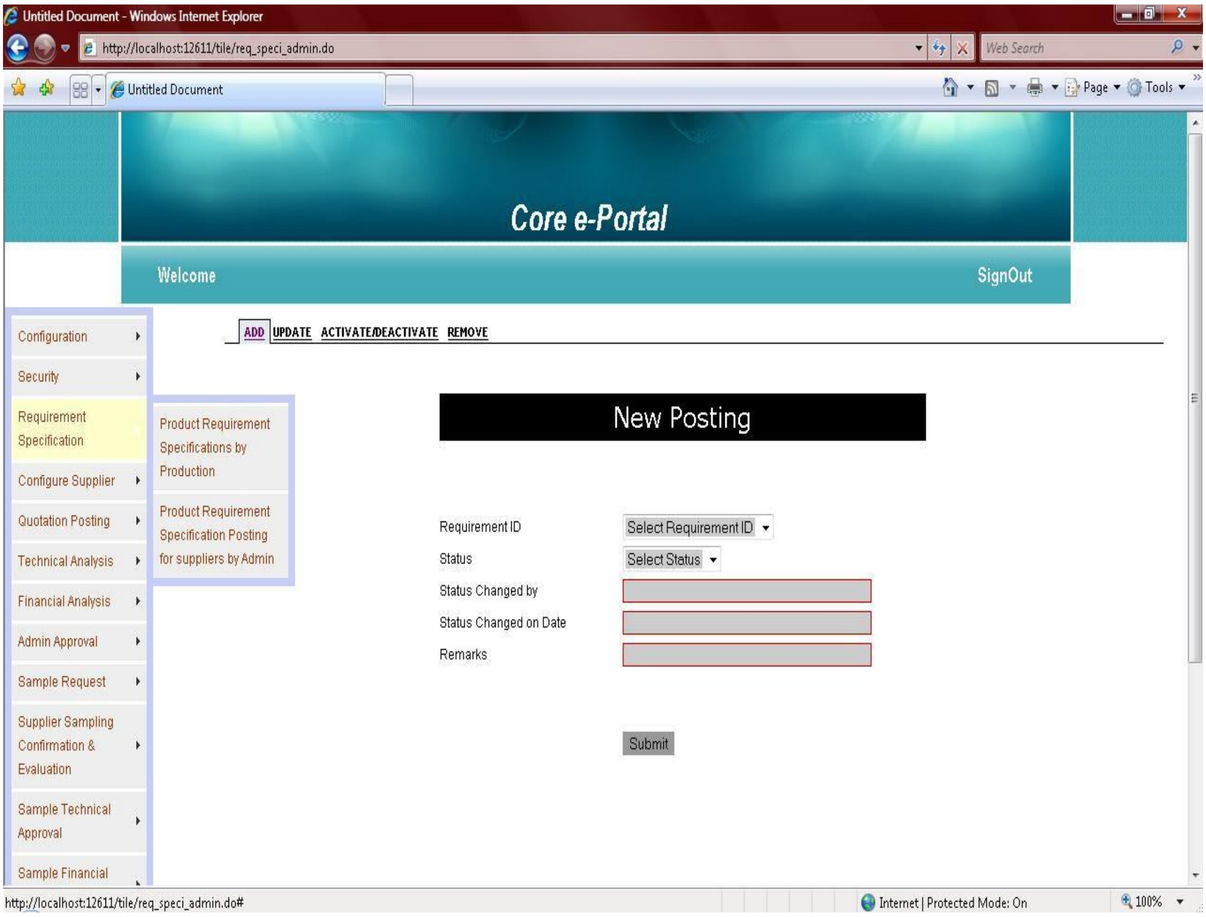


Add Specification

Material ID	Select Material ID
Quantity	
Unit	Select
Request Posting Opening Date	
Request Posting Closing Date	
Posted by	
Priority	Select

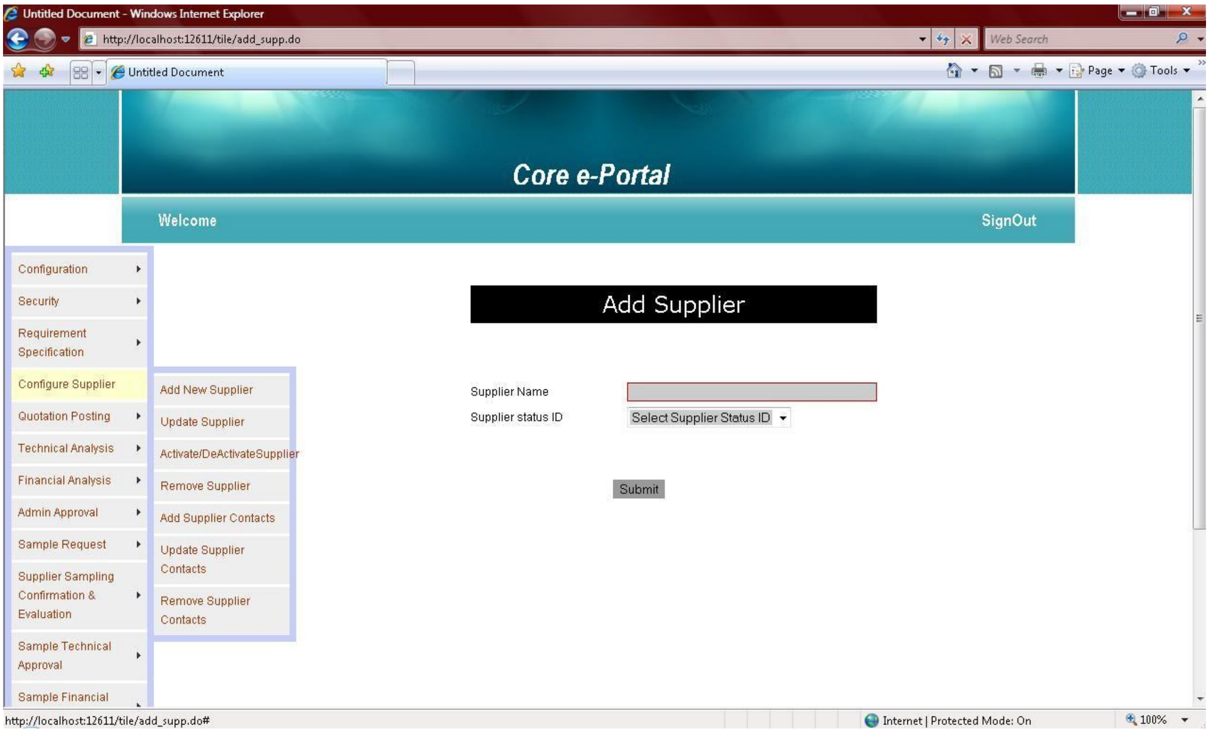
Requirement specification by Production

The details of the requirements are entered by the production department from this part. The suppliers view these requirements and post there quotations accordingly.


Untitled Document - Windows Internet Explorer
http://localhost:12611/tile/req_speci_admin.do
Web Search
Untitled Document
Page
Tools
Core e-Portal
Welcome
SignOut
Configuration
Security
Requirement Specification
Configure Supplier
Quotation Posting
Technical Analysis
Financial Analysis
Admin Approval
Sample Request
Supplier Sampling Confirmation & Evaluation
Sample Technical Approval
Sample Financial
Product Requirement Specifications by Production
Product Requirement Specification Posting for suppliers by Admin
ADD
UPDATE
ACTIVATE/DEACTIVATE
REMOVE
New Posting
Requirement ID
Select Requirement ID
Status
Select Status
Status Changed by
Status Changed on Date
Remarks
Submit
http://localhost:12611/tile/req_speci_admin.do#
Internet | Protected Mode: On
100%

Requirement Specification posting by Admin

The admin posts the requirements for the suppliers who are posting their quotations. These posting can also be updated or removed as the need may be.



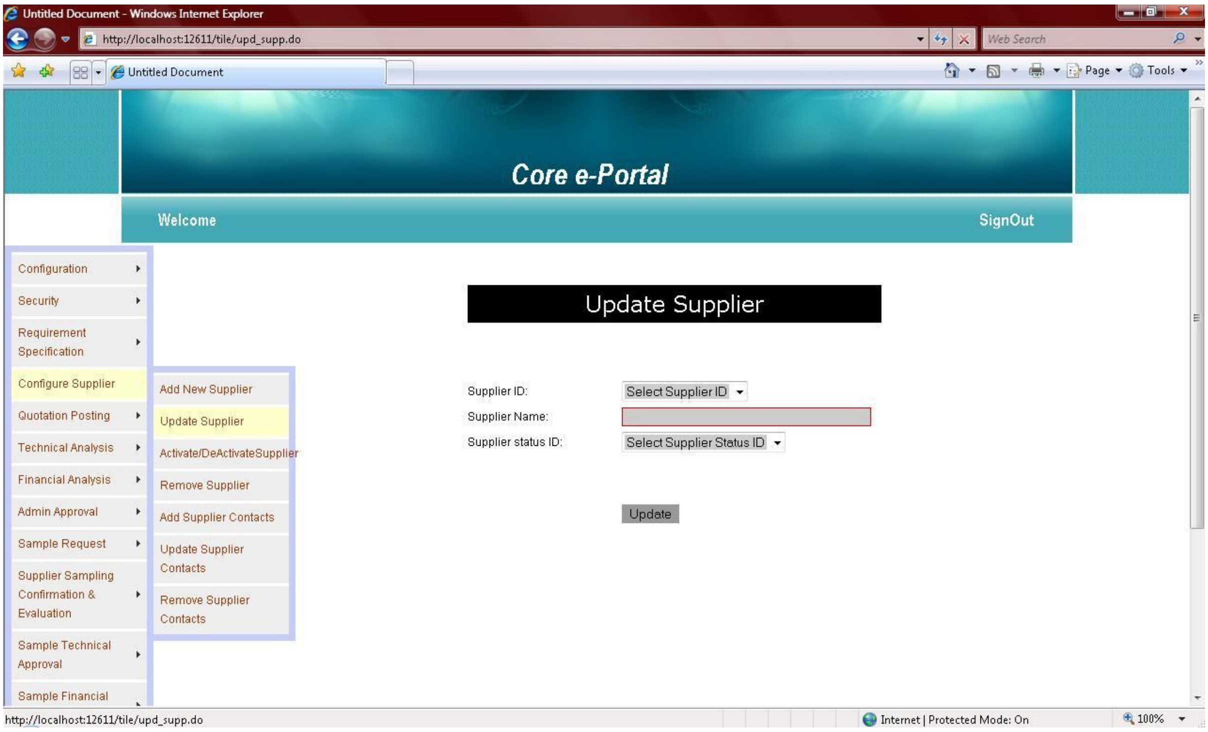
Add Supplier

Supplier Name

Supplier status ID: Select Supplier Status ID

Submit

New suppliers are added for posting quotations using this menu.



Update Supplier

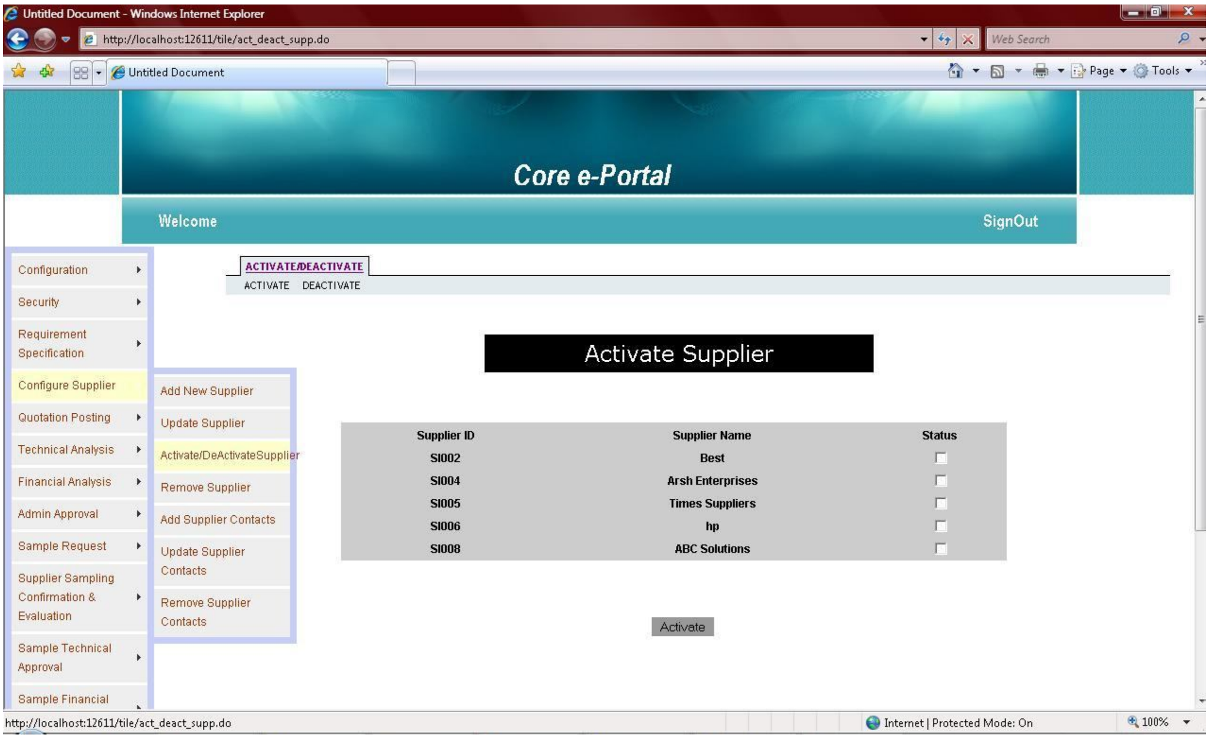
Supplier ID: Select Supplier ID

Supplier Name:

Supplier status ID: Select Supplier Status ID

Update

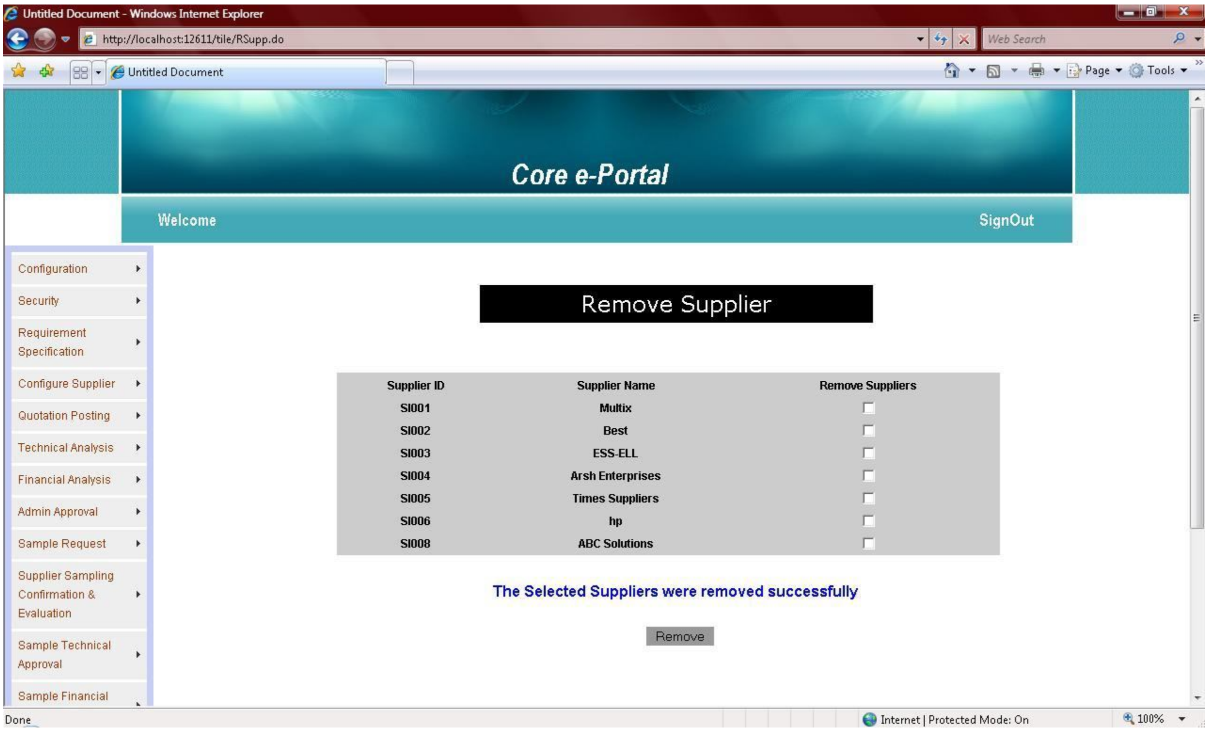
The supplier details can be updated from this part.



Activate Supplier

Supplier ID	Supplier Name	Status
SI002	Best	☐
SI004	Arsh Enterprises	☐
SI005	Times Suppliers	☐
SI006	hp	☐
SI008	ABC Solutions	☐

The status of supplier is changed from here.

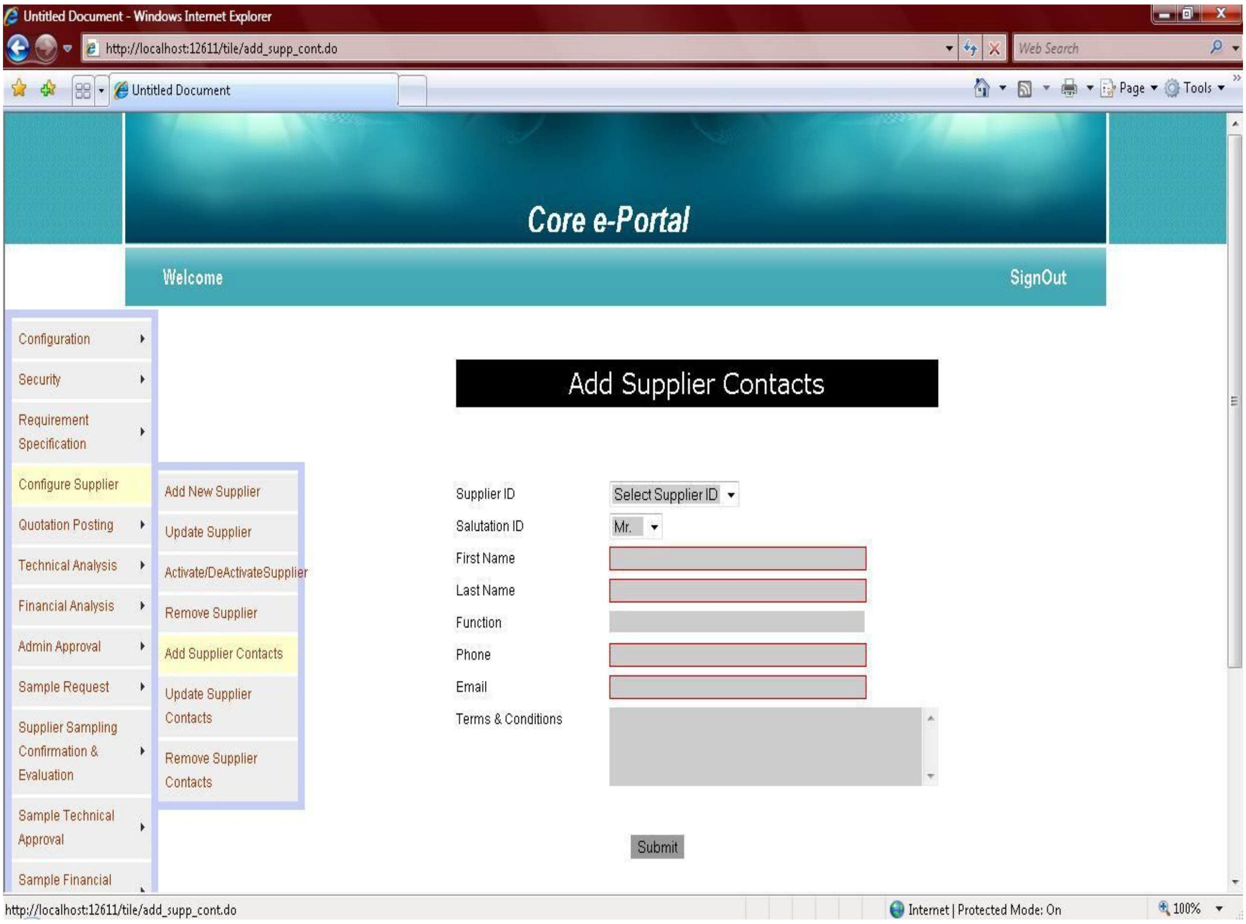

Untitled Document - Windows Internet Explorer
http://localhost:12611/tile/RSupp.do
Core e-Portal
Welcome
SignOut
Configuration
Security
Requirement Specification
Configure Supplier
Quotation Posting
Technical Analysis
Financial Analysis
Admin Approval
Sample Request
Supplier Sampling Confirmation & Evaluation
Sample Technical Approval
Sample Financial
Remove
Done
Internet | Protected Mode: On
100%

Remove Supplier

Supplier ID	Supplier Name	Remove Suppliers
SI001	Multix	☐
SI002	Best	☐
SI003	ESS-ELL	☐
SI004	Arsh Enterprises	☐
SI005	Times Suppliers	☐
SI006	hp	☐
SI008	ABC Solutions	☐

The Selected Suppliers were removed successfully

The suppliers can be removed from the bidding from this part.

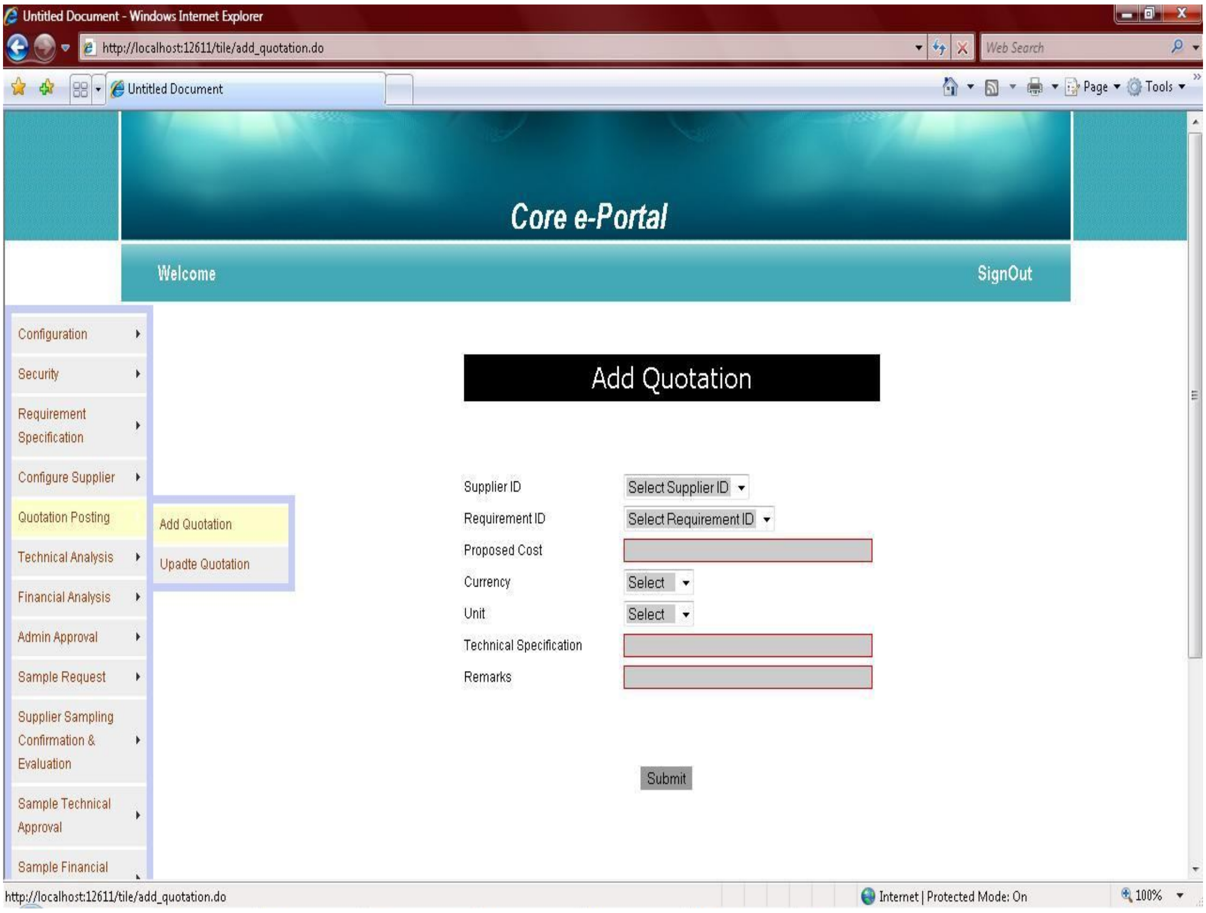


Add Supplier Contacts

Supplier ID	Select Supplier ID
Salutation ID	Mr.
First Name	
Last Name	
Function	
Phone	
Email	
Terms & Conditions	

Supplier Contacts

The supplier’s contact details are to be entered for further reference. Details are added updated or deleted using this part.



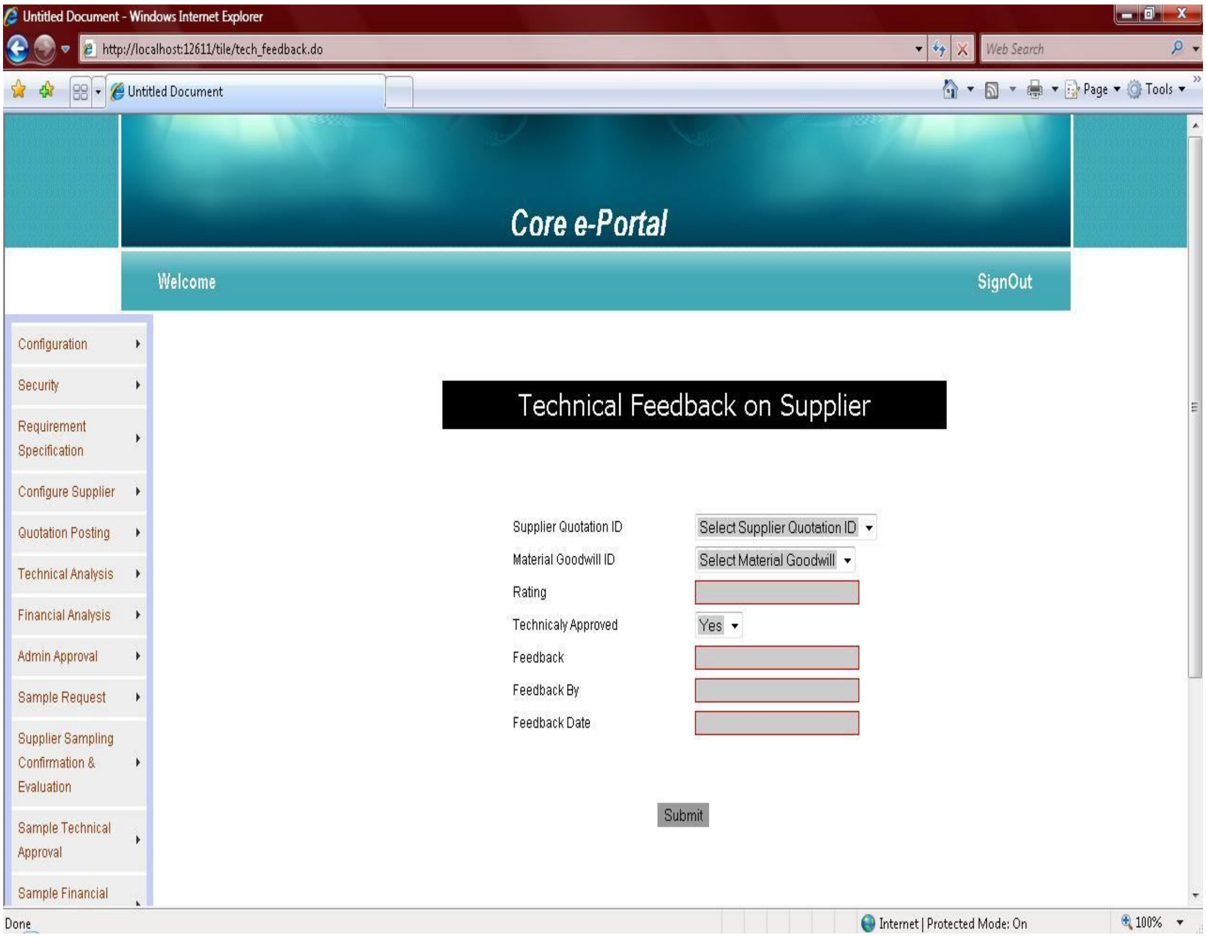
Add Quotation

Supplier ID	Select Supplier ID
Requirement ID	Select Requirement ID
Proposed Cost	
Currency	Select
Unit	Select
Technical Specification	
Remarks	

Submit

Quotation Posting

The quotations are posted by the suppliers for the bidding process. They can be updated before the closing date for the bid.

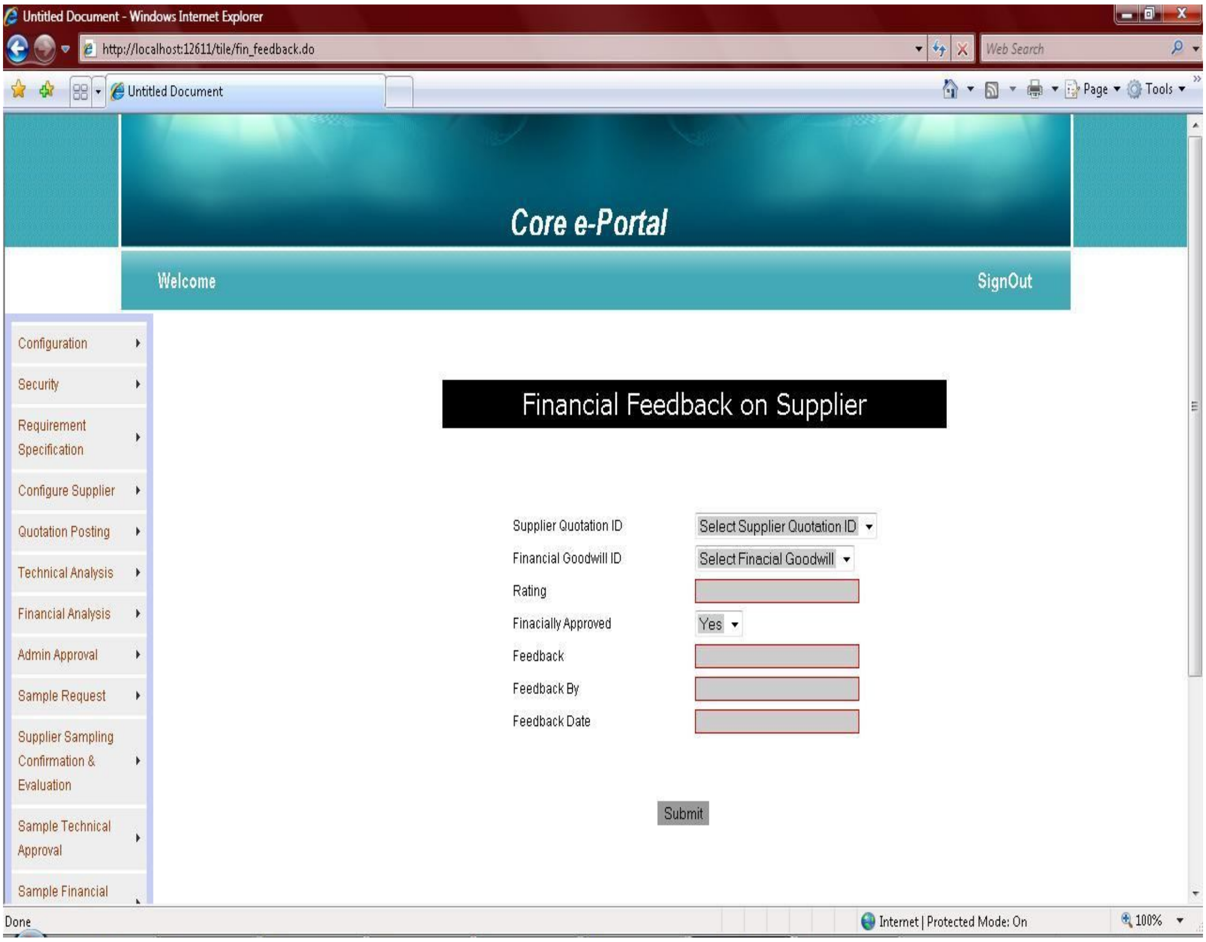


Technical Feedback on Supplier

Supplier Quotation ID	Select Supplier Quotation ID
Material Goodwill ID	Select Material Goodwill
Rating	
Technicaly Approved	Yes
Feedback	
Feedback By	
Feedback Date	

Technical Analysis

The admin views the quotations of all the suppliers and rates them based on their technical feasibility owing to the requirements.

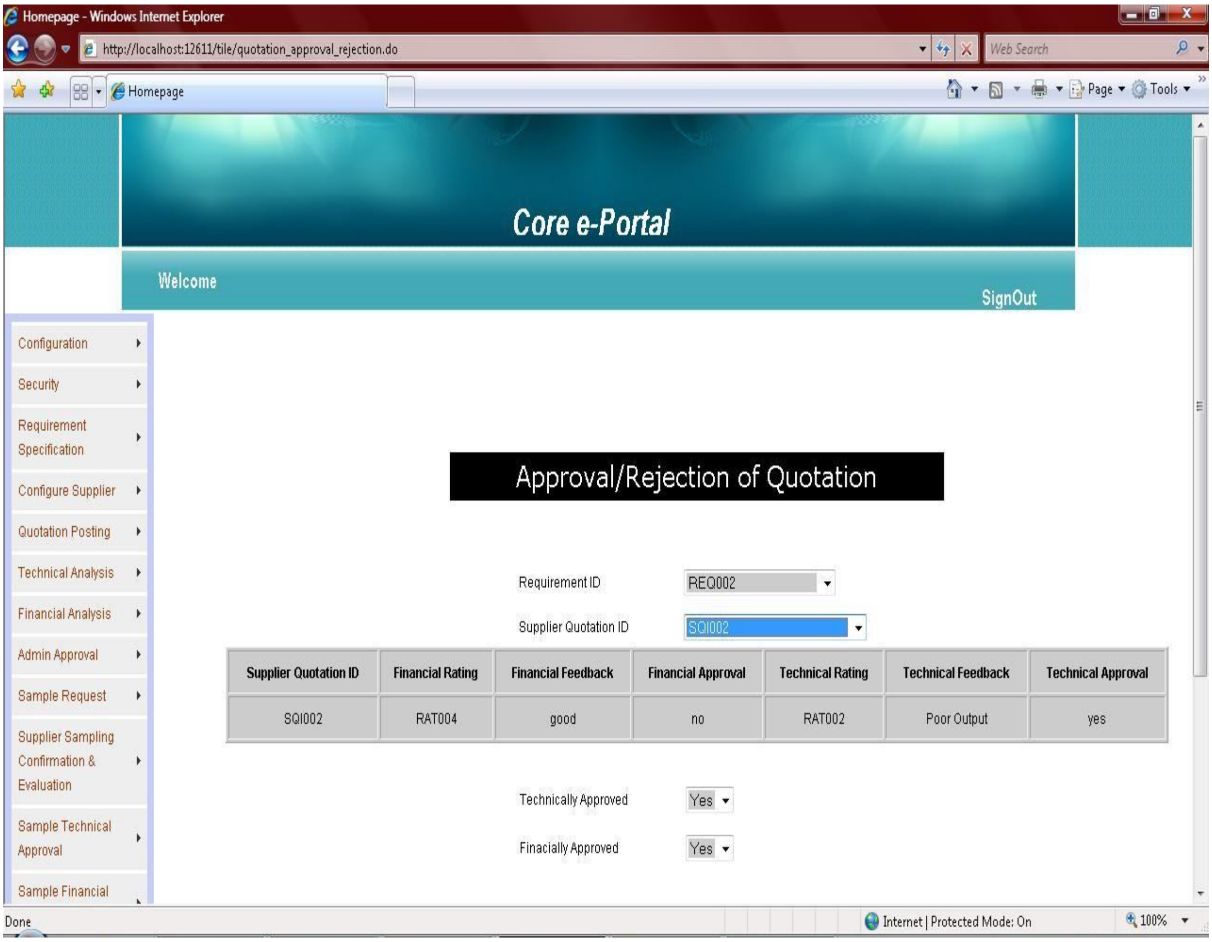


Financial Feedback on Supplier

Supplier Quotation ID	Select Supplier Quotation ID
Financial Goodwill ID	Select Finacial Goodwill
Rating	
Finacially Approved	Yes
Feedback	
Feedback By	
Feedback Date	

Financial Analysis

The admin views the quotations of all the suppliers and rates them based on their financal feasibility owing to the requirements


Homepage - Windows Internet Explorer
http://localhost:12611/tile/quotation_approval_rejection.do
Web Search
Homepage
Page
Tools
Core e-Portal
Welcome
SignOut
Configuration
Security
Requirement Specification
Configure Supplier
Quotation Posting
Technical Analysis
Financial Analysis
Admin Approval
Sample Request
Supplier Sampling Confirmation & Evaluation
Sample Technical Approval
Sample Financial
Done
Internet | Protected Mode: On
100%

Approval/Rejection of Quotation

Requirement ID: REQ002

Supplier Quotation ID: SQI002

Supplier Quotation ID	Financial Rating	Financial Feedback	Financial Approval	Technical Rating	Technical Feedback	Technical Approval
SQI002	RAT004	good	no	RAT002	Poor Output	yes

Technically Approved: Yes

Finacially Approved: Yes

Admin Approval

The admin approves the quotation for sample after it has been both technically and financially approved.

7. TESTING

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

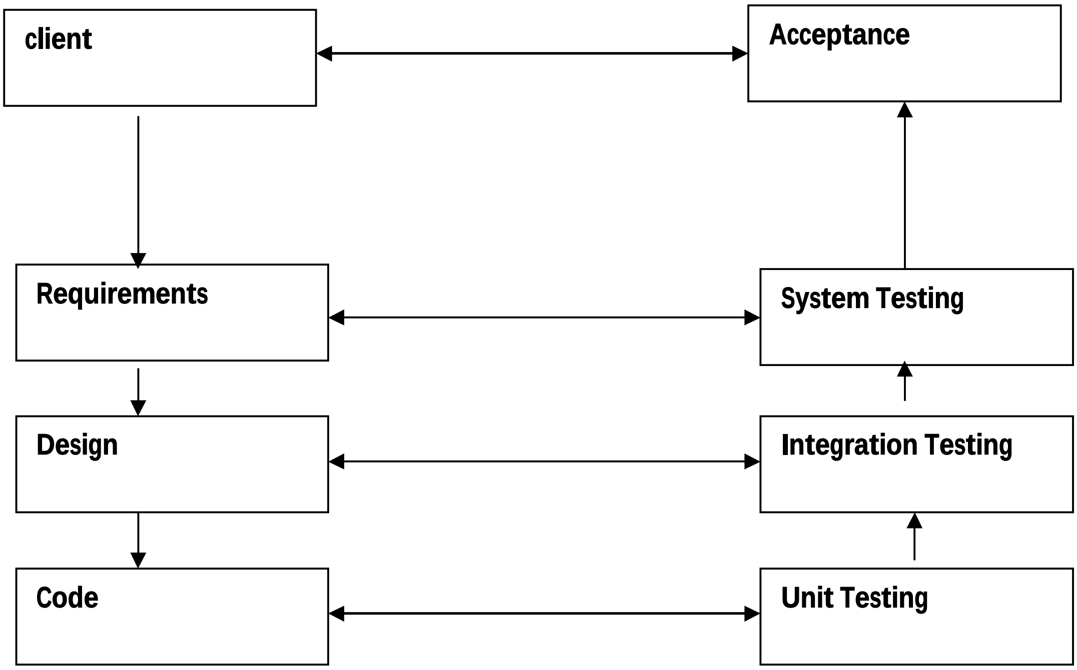
A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing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

Objective of Testing

A good testing case is one that has high probability of finding as yet undiscovered errors. Testing is a process of executing a program with the intent of finding errors in it .A successful test is one that uncovers an as yet undiscovered error .The aim of testing is to determine that a program works by showing that it has no errors .

Levels of Testing:-

Testing is usually relying on to detect the faults on each phase, in addition to the faults introduced during the coding phase itself. Due to this, different levels of testing are used in the testing process, each level of testing aims to test different aspect of a system.



STRATEGIC APPROACH TO SOFTWARE TESTING

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

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

Table Given below outlines the tests that were performed on the system to ensure correctness and unearth errors, which were subsequently debugged.

Table shows the Tests Conducted on the System

Testing Phase	Objectives
Unit Testing	The various functions within each program and the program blocks are tested for proper working.
Module Testing	A module is composed of various programs related to that module. Module testing is done to check the module functionality and interaction between units within a module
Integration Testing	Integration testing is done to test the functionality and interfacing between the modules.
Acceptance Testing	Acceptance testing is done after implementation to check if the system runs successfully in the customer environment/site.

UNIT TESTING

Unit Testing will be done to test field validations, navigation, functionality of the programs and its blocks. These tests are applied on various functions within each program and other critical program blocks. Table given below gives the outline of three-sample test cases for Unit Testing performed on the system.

MODULE TESTING

Module testing will be done to test the interaction between the various programs within one module. It checks the functionality of each program with relation to other programs within the same module. It then tests the overall functionality of each module. Tables given below outline two sample test cases for Module Testing performed on the system.

INTEGRATION TESTING

Integration testing is done to test the functionality and interfacing between the modules. The system is built up of various modules, which work together to automate the activities of the HR department. These modules should work together in a seamless way to achieve the desired results. Integration testing will test for this property of the modules. The modules display a cause and effect relationship, if data in one module is changed, then it affects the data to change in some other module also. Integration testing needs to check if the modifications do not adversely affect some other modules.

ACCEPTANCE TESTING

Acceptance testing was done after the implementation of the system. The acceptance testing will check if the system works correctly in the user environment and if the entire user specified functionalities are present. It also tests if the system adheres to the company policies and quality standard. The HR Database system was tested and accepted by Sierra Atlantic after the acceptance testing.

VALIDATION CHECKS

Following type of checks/ data validations have been used:

- Data type
- Length
- Constraints
- Blank field
- Format

Data type:

I have use String type for character, int for numeric, and Date for date type. No numeric field insert in date. Character never inputted in numeric field as phone no never accept character if any person input wrongly give message. When this problem is removed then user performs further operation.

Length:

When we define a max length. Then it never accepts more data .for example if I define numeric length is 5 then it store either equal to length or less than length. If user gives more character than required then display message and stop processing.

Constraints:

In this I am defining range of data if data is less than then display error with message. For example code of password is four characters. The field of date must be 8 characters.

Blank field:

When users add data and some field is blank then it display message with out halt, But stop processing.

Format:

The pre define format is used not change daily to daily for example format of date DDMMYYYY: 01012002 is used in all date type field. If user inserts an other format then display message.

Web Design Constraints

The following design constraints were kept in mind while designing the pages for the whole application**:**

The pages should be consistent and easy to operate. It should be designed in such a way that an average user who does not have much idea about JSP and related technology can still be able to access the information needed. The navigation should be easy and stepwise .A customer may have multiple accounts and should be able to access all the accounts with a single password .a request number should be generated so that it is easy to refer to it at a later date. The entire official should work as administrators and should have access to all the service areas / pages of the application by the transaction areas / pages are accessible to only customers whose accounts are not frozen.

DESIGN

With the above constraints in mind, we have designed an interface for the pages that is consisting of three different components namely:

Header

Header contains information about the company and the logo and like.

Navigation Panel

Navigation panel has all the options available to the user. At any point of time, any of these options can be selected and used. All these are hyperlink and connect to different JSP files used.

Main working panel

Main working panel is the area on the screen that is used by the application to allow user to interact with it. There are some pages where some relative actions must be taken. These actions are denoted as buttons so that it stands – out from the hyperlink and are easy to use.

To accomplish the state structure two framed pages are used – one for the customers and another one for the administrator. As the two types of users have different set of operations to perform in the system. it demanded two separate pages for the purpose . These frames are called depending on the login process.

8. IMPLEMENTATION

A crucial phase in system development is the successful implementation of the new system design. Implementation includes all those activities that take place to convert from the old system to the new system to the new system. The new system may be completely new replacing an existing manual or automated system or it may be major modification to an existing system.

In either case proper implementation becomes necessary so that a reliable system based on the requirement of the organization can be provided. Successful implementation may not guarantee improvement in the organization using the new system, but improper installation will prevent this improvement.

It has been observed that even the best system cannot show good results if the analysts managing the implementation do not attend every important detail. This is the area where the system analysts need to work with utmost care.

There are three main aspects of implementation:-

- User training
- User Manual
- Conversion

User Training

Even well designed system can succeed or fail because of the way these are operated and used. Therefore the quality of training received by the personal involved with the system in various capacities helps or hinders and may even prevent the with the system development must know in detail what their roles will he , how they can make efficient use of the system and what the will not do for them.

An analysis of user training focuses on two factors:-

User capabilities

Nature of the proposed system

When a system is developed then only the developer of it knows its operation. But as the project is being made for a client (organization) and it is to be used by the employees of that firm, so there is need of user training. As the system is new for the employees and staff of the cooperation so there is a necessity to train them because they have to operate the system. For the live demonstration with personal contact are very effective. No training is complete without familiarizing users with simple system

User Manual:-

When a system is developed then only the developer of it knows its operation. But as the project is being made for a client and it is to be used by the employees of that firm, so there is a need of user manual. It is documentation of the project that describes how to operate the system? What are the various modules for efficient use of the system? How to start the system?

The user manual helps to find out the answers of the queries like:-

How to start the system?

How to make entry?

How to generate reports?

The System being developed is web designing application and can be run simply in the Web browser…

Conversion

Conversion is the process of from the old system to the new one. It should be accomplished in shortest possible time. Four methods are commonly used –

Parallel system

Direct conversion

Pilot system

Phase – in mind

Phase – in mind

The **Core ePortal-SRFQ** is being developed with different phases being taken into mind…the project is developed in different phases starting with the analysis phase and ends with implementation phase. After implementation phase the maintenance phase will begin which keeps on going….

9. SYSTEM SECURITY

CHECKS AND CONSTRAINTS

System security refers to various validations on data in form of checks and controls to avoid the system from failing. It is always important to ensure that only valid data is entered and only valid operations are performed on the system. The system employs two types of checks and controls:

Client Side Security

Various client side validations are used to ensure on the client side that only valid data is entered. Client side validation saves server time and load to handle invalid data. Some checks imposed are:

A JavaScript program is used to fill up date in the date fields only.

JavaScript is used to ensure that numeric field is filled with numeric data only.

Maximum lengths of the fields of the forms are appropriately defined.

Forms can not be submitted without filling up the mandatory data so that manual mistakes of submitting empty fields that are mandatory can be sorted out at the client side to save the server time and load.

Tab-Indexes are set according to the need and taking into account the ease of user while working with the system.

Server Side Security

Some checks can’t be applied at client side. Server side checks are necessary to save the system from failing and intimating the user that some invalid operation has been performed or the performed operation is restricted. Some of the server side checks imposed are:

Server side constraint has been imposed to check for the validity of Primary key and foreign key. A primary key value cannot be duplicated. Any attempt to duplicate the primary key value results into a message intimating the user about that. Values through the forms using foreign key can be updated only for the existing foreign key values.

Exceptions are caught in the programs and are used constructively to avoid system failure. User is intimated through appropriate messages about the successful operations or exceptions occurring at server side.

Access permissions to various types of users are controlled according to the organizational structure. Only permitted users can log on to the system and can have access according to their category. User-name, passwords and permissions are controlled on the server side.

Using Oracle as back-end provides best security features of database.

10. CONCLUSIONS

This was the first considerably large and important project undertaken by me during my B.Tech course. It was an experience that changed the way I perceived project development. The coding could not be started before the whole system was completely finalized. Even then there were so many changes required and the coding needed to be changed. I attribute this to inadequate information gathering from the user. Though there were many meetings with the user and most of the requirements were gathered, a few misinterpretations of the requirements still crept in. It made me realize how important the systems analysis phase is. The project is a classic example for the adage that learning of concepts needs to be supplemented with application of that knowledge.

On the whole it was a wonderful experience developing this project and I would have considered my education incomplete without undertaking such a project which allowed me to apply all that I have learnt.

10.1 Merits of the project

The project is identified by the merits of the system offered to the user. The merits of this project are as follows –

- This project offers user to enter the data through simple & interactive forms. This is very helpful for the client to enter the desired information through so much simplicity.
- The user is mainly more concerned about the validity of the data, whatever he is entering. There are checks in every stage of any new creation, data entry or updating so that the user cannot enter the invalid data, which can cause problems at a later date.
- Sometimes the user finds in the later stages of using the project that he needs to update some of the information that he entered earlier. There are options for him by which he can update the records. Moreover there is restriction for him that he cannot change the primary data field. This keeps the validity of the data to longer extent.
- From every part of the project the user is provided with the links through framing so that he can go from one option of the project to other as per the requirement. This is bound to be simple and very friendly as per the user is concerned. That is, we can say that the project is user friendly which is one of the primary concerns of any good project.
- Data storage and Retrieval will become faster and easier to maintain because data is stored in a systematic manner and in a single database.
- Decision making process would be greatly enhanced because of faster processing of information since data collection from information available on computer takes much less time then manual system.

- Allocation of Sample Results becomes much faster because at a time the user can see the records of last years.
- Generation of Acknowledgement, Dispatches and Sample Reports and any type of report will be much quicker now than done manually and with out much paper handling.
- Through these features it will increase the efficiency, accuracy and transparency.
- Since the Computers would be networked together, information can travel from one department to other Department and back very quickly. Thus communication will be rapid.

10.2 Demerits of the project

The project is tried up to best to provide the simplicity, capability and reliability to the user but then also there are fields, which cannot be tackled up to best of the results. This project also has some of the demerits in it. As nothing is perfect in this world, we also not claim that our software is perfect. As all other, our project also has some demerits.

Those demerits are as follows –

- The user has not provided any option to delete any record, which he has entered once.
- Due to the maintenance of data related with the employee who had either resigned or retired or currently not working in company, the size of the database becomes too bulky which leads the processing speed of any request from the client to become a bit slow.

11. FUTURE SCOPE OF THE PROJECT

It is unreasonable to consider a computer-based information system complete or finished; the system continues to evolve throughout its life cycle, even if it’s successful. Due to the creative nature of the design, there remain some lapses inaccurate communications between the users and the developers. So, certain aspects of the system must be modified as operational experience is gained with it. As users work with the system, they develop ideas for change and enhancements.

12. REFERENCE

1. Complete Reference for JAVA.
2. Complete Reference for J2EE.
3. Alam, M. (2011). Online Banking (1st ed.). New Delhi: LAP LAMBERT Academic Publishing, ISBN: 978-620-3-86302-4. DOI: 10.6084/m9.figshare.14612127
4. Alam, M. (2012). Electronic Ticket Machine (1st ed.). New Delhi: LAP LAMBERT Academic Publishing, ISBN: 978-620-3-86332-1, DOI: 10.6084/m9.figshare.14661354
5. Alam, M. (2013). Just Shop-Shopping (1st ed.). New Delhi: LAP LAMBERT Academic Publishing, ISBN: 978-620-3-58124-9, DOI: 10.6084/m9.figshare.14431382
6. www.google.com
7. www.stackoverflow.com
8. www.wikipedia.com