**SYSTEM REQUIREMENT SPECIFICATION**

**OVERVIEW**

The motto of this project is information transparency. Because of this, a common citizen can also play a vital role in controlling the corruption

**STUDY OF THE SYSTEM**

In the flexibility of uses the interface has been developed a graphics concepts in mind, associated through a browser interface. The GUI’s at the top level has been categorized as follows

1. Administrator Interface Design.
2. User Interface.
3. Security Authentication.
4. Reports.
5. General end-users.

The administrative user interface will maintain the different users details, the interface helps the administration with all the transactional states like which users sending the mails, and which users receiving whishing mails, users details information history. And the statistics of the system in difference strategies.

**NUMBER OF MODULES**

The system after careful analysis has been identified to be presented with the following modules:

1. Budget Officer (Administrator )

2. Department Budget Officer

3. District Budget Officer

4. Contractor

5. Citizen

**Module Description:**

1. Budget Officer:

Budget Officer (Administrator) can add new departments and add sub departments under departments. Administrator can add department Officers, district Officers and view department and district officers list, and can delete officer also. Budget Officer can enter state budget of a particular financial year at department level and sub department level and can modify, delete as per requirement. Administrator also gives replies to queries.

2. Department Budget Officer:

Department Officer can view department wise budget details of our state budget. Department officer can make request to officer to get fund of particular department. Department officer can make transactions through secure transaction id. Then Department officer can distributed fund to districts under concern departments. Can view distributed Budget and send queries to administrator.

1. District Budget Officer:

District Budget officer can make request to Department officer to get fund of particular department. District officer can make transactions through secure transaction id. Then District Officer can able to distributed fund to different works at districts level under concern departments. And District officers can select contractors for particular works also.

1. Contractors :

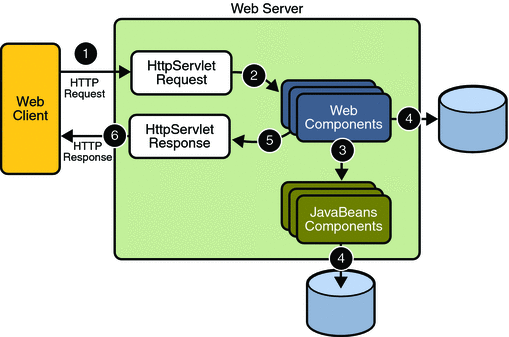
Contractors can view work details, make request to get money. And make transaction using transactions password. Contractor can enter work details. They can update their work details with current status also. Contractors also can send queries to budget officer.

1. Citizen:

Citizen can view state budget each and every department wise and district budget. Citizen can view work details of particular department and districts wise also. Citizen can send queries to officer regards to works. Citizens have facilities to give feedback

**PROCESS FLOW**

**ARCHITECTURE DIAGRAM**



1. **THE PRESENTATION LAYER**

Also called as the client layer comprises of components that are dedicated to presenting the data to the user. For example: Windows/Web Forms and buttons, edit boxes, Text boxes, labels, grids, etc.

1. **THE BUSINESS RULES LAYER**

This layer encapsulates the Business rules or the business logic of the encapsulations. To have a separate layer for business logic is of a great advantage. This is because any changes in Business Rules can be easily handled in this layer. As long as the interface between the layers remains the same, any changes to the functionality/processing logic in this layer can be made without impacting the others. A lot of client-server apps failed to implement successfully as changing the business logic was a painful process

1. **THE DATA ACCESS LAYER**

This layer comprises of components that help in accessing the Database. If used in the right way, this layer provides a level of abstraction for the database structures. Simply put changes made to the database, tables, etc do not affect the rest of the application because of the Data Access layer. The different application layers send the data requests to this layer and receive the response from this layer.

1. **THE DATABASE LAYER**

This layer comprises of the Database Components such as DB Files, Tables, Views, etc. The Actual database could be created using SQL Server, Oracle, Flat files, etc.   
In an n-tier application, the entire application can be implemented in such a way that it is independent of the actual Database. For instance, you could change the Database Location with minimal changes to Data Access Layer. The rest of the Application should remain unaffected.

**SDLC METHODOLOGIES**

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, “A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

The steps for Spiral Model can be generalized as follows:

* The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
* A preliminary design is created for the new system.
* A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
* A second prototype is evolved by a fourfold procedure:

1. Evaluating the first prototype in terms of its strengths, weakness, and risks.
2. Defining the requirements of the second prototype.
3. Planning an designing the second prototype.
4. Constructing and testing the second prototype.

* At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involved development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer’s judgment, result in a less-than-satisfactory final product.
* The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
* The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
* The final system is constructed, based on the refined prototype.
* The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.

**The following diagram shows how a spiral model acts like:**



**Fig 1.0-Spiral Model**

**ADVANTAGES**

* Estimates(i.e. budget, schedule etc .) become more relistic as work progresses, because important issues discoved earlier.
* It is more able to cope with the changes that are software development generally entails.
* Software engineers can get their hands in and start woring on the core of a project earlier.

**SOFTWARE REQUIREMENT AND**

**HARDWARE REQUIREMENT**

**SOFTWARE REQUIREMENTS**

Operating System : Windows XP/2003 or Linux

User Interface : HTML, CSS

Client-side Scripting : JavaScript

Programming Language : Java

Web Applications : JDBC, Servlets, JSP

IDE/Workbench : My Eclipse 6.0

Database : Oracle 10g

Server Deployment : Tomcat 6.x

**HARDWARE REQUIREMENTS**

Processor : Pentium IV

Hard Disk : 40GB

RAM : 1GB or more