FUNCTIONAL SPECIFICATION

Project Code:	DMS-01
Project Name:	Directory Monitoring System

25-Apr-2011		
Prepared by/Date	**Reviewed by/Date**	**Approved by/Date**

Revision History

Version (x.yy)	Date of Revision	Description of Change	Reason for Change	Affected Sections	Approved By
1.00	25-Apr-2011	Initial Draft			

Affected Groups

Development Engineering
Quality Assurance
ABC Ltd

List of Reference Documents

Name	Version No.
1. Request For Proposal	1.2
2.	
3.	
4.	

Table of Contents

1. Introduction

1.1 Background
ABC is a company which runs its application software on UNIX systems.

1.2 Purpose
It is proposed to develop a system to Directory Monitoring System which helps users to keep track of changes made to files and directories in the UNIX file system.

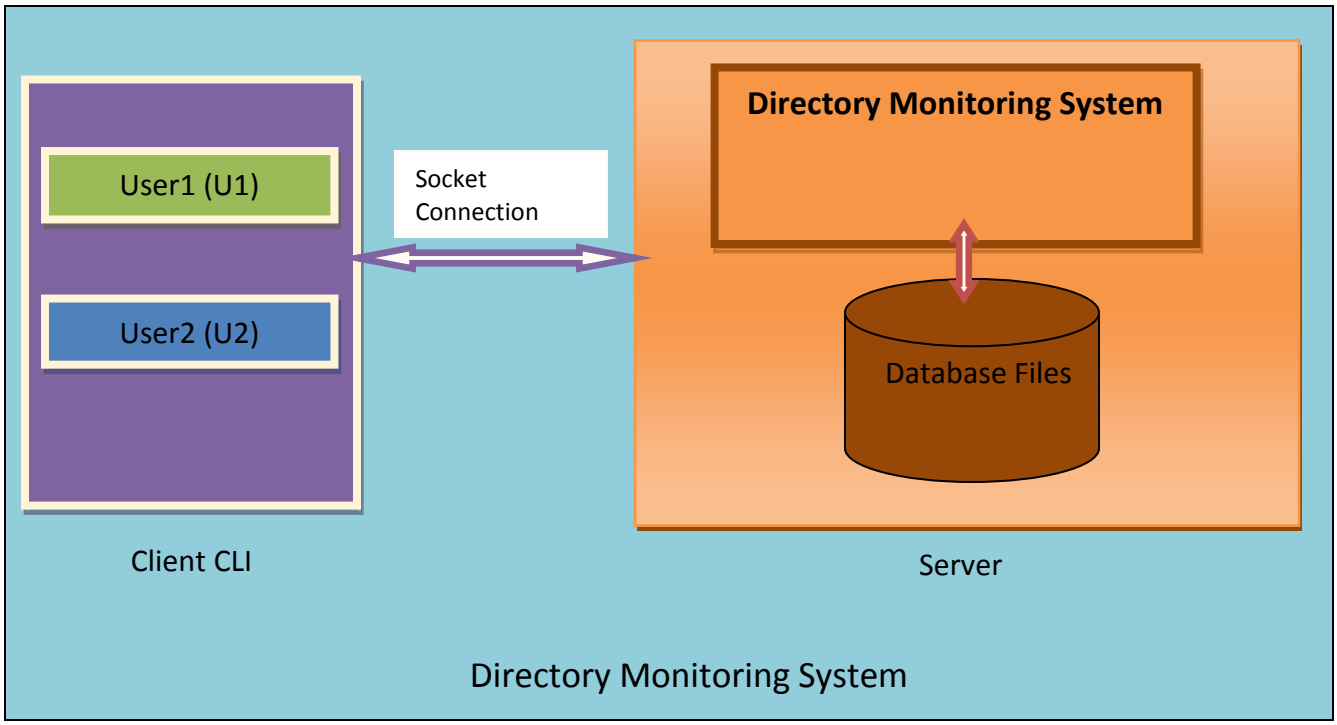
1.3 Scope
The scope of the Directory Monitoring System will be to provide the functionality as described in Functional Requirements below. The system will be developed on a Linux box using C language and would provide a console-based user-interface.

2. Systems Overview

2.1 *System Description*
The Directory Monitoring System should support the following users.

- User1 (U1)
- User2 (U2)

The functional flow of the messages across different application components is shown below.



2.2 Environment

The system will be developed on a Linux box using C language and would provide a console-based user-interface.

- Intel hardware machine (PC P4-2.26 GHz, 512 MB RAM, 40 GB HDD)
- Linux 2.6 or higher
- Compilers – gcc or g++

3. Sub-system Details

3.1 User 1 and User 2

The Users should be able to do the following operations:

- Request for a file/directory to be monitored
- View status of files/directories being monitored
- Place multiple requests from the same client on different files/directories.
- Log all changes to the files/directories being monitored in separate log file per client
- Specify maximum number of log files in a config file
- Size of each log file should be configurable in a config file

Request for a file/directory to be monitored: As part of this operation, the user must be able to request a specific file or directory to be monitored.

(Refer to section 4. Data Organization –> Table1: Monitoring Information)

View status of files/directories being monitored: As part of this operation, the user must be able to request for the status of a specific file or directory being monitored.

Place multiple requests from the same client on different files/directories: As part of this operation, the user must be able to place request to monitor multiple files or directories from the same client.

Log all changes to the files/directories being monitored in separate log file per client: As part of this operation the DMS should log all changes made to files/directories being monitored in a separate log file per client.

Specify maximum number of log files in a config file: Changes to files and directories being monitored are logged into log files. The maximum number of log files is a configurable value which is present in a config file.

Size of each log file should be configurable in a config file: The size of each log file is also a configurable value which is present in a config file.

4. Data Organization

The log files are stored in **/tmp/monitor/** folder. The “monitor” folder has to be created the first time when the server program starts.

This section explains the data storage requirements of Directory Monitoring System application and ***indicative*** table (database) structure. The following sections explain few of the tables required for the application and the other tables will have to be designed accordingly.

4.1 Table: Monitoring Information

The Server on request from the user gives details of clients, file/directory being monitored and log file name for each.

Table 1: Monitoring Information

Field Name	Data Type	Description
ClientNo	int	Specifies the number of clients connected to the server
MonitorData	char[100]	Name of file/directory being monitored
LogFileName	char *	Specifies name of Log File Dynamically create new "*LogFileName*" every time a client connects the server

Note: Similarly, create other necessary normalized tables.

5. Assumptions

- The file/directory is monitored on Unix system
- Both Server and Client process run on the same system

6. Expectations

- The application should be designed using client server architecture
- The server should be a concurrent server servicing multiple clients
- Compilation and Build should be done using makefile
- Source-code and all documents must be maintained (checked-in) in configuration management system (subversion)
- Wipro's coding standards (for C/C++) should be followed
- Deliverables should include use-case diagrams, design document, compiled and tested source code, test-plans, test-cases documents, test-results and release note

7. Acceptance Criteria

All P1 requirements has to be mandatorily implemented

8. Traceability to Requirements

Document Reference ID & Description: (Doc ID from which this document is derived)		
Sl. No.	**Reference document: RS Requirement/Feature (Section ID/Name)**	**Current document: FS Location (Section ID/Name)**
1.	Requirements as mentioned in Section 2.1	Section 3.1, 4.1
2.	Requirements as mentioned in Section 2.2	Section 3.1, 4.1
3.	Requirements as mentioned in Section 2.3	Section 3.1

9. Acronyms and Glossary

Abbreviation	Remark
DMS	Directory Monitoring System
RS	Requirement Specification
FS	Functional Specification