Software Requirements Specification

DMT/RE10/TMP

|  |  |
| --- | --- |
| **Project Code** | Systems-1520 |
| **Project Name** | FILE SYSTEM |

|  |  |  |
| --- | --- | --- |
| **Prepared/Modified by** | **Role** | **Date of Preparation** |
| Sneha Nair | Developer | 31/12/2013 |
| **Reviewed by** | **Role** | **Date of Review** |
| Swapnil Kadam | Team Leader | 07/01/2014 |
| **Approved by** | **Role** | **Date of Approval** |
|  | Project Mentor |  |
| **Circulation List** |  | **Version Number of the template:1.0** |
| **Version Number** | 1.1 |  |

Table of Contents

[1. Introduction 3](#_Toc371689802)

[1.1 Purpose 3](#_Toc371689803)

[1.2 Document Conventions 3](#_Toc371689804)

[1.3 Definitions, Acronyms and Abbreviations 3](#_Toc371689805)

[1.4 References 4](#_Toc371689806)

[1.5 Process for Requirements Capturing 4](#_Toc371689807)

[2. Project Scope 5](#_Toc371689808)

[2.1 Scope 5](#_Toc371689809)

[2.2 Out of Scope 5](#_Toc371689810)

[3. Project Description 6](#_Toc371689815)

[3.1 Product Perspective 6](#_Toc371689816)

[3.2 Product Function / Business Process Overview 6](#_Toc371689817)

[3.4 Assumptions and Dependencies 6](#_Toc371689819)

[4. Functional Requirements 7](#_Toc371689820)

[4.1 Functionality 7](#_Toc371689822)

[5. Technical Requirements 10](#_Toc371689823)

# Introduction

## Purpose

The purpose of this project is to develop File system. File system provides layer of abstraction so that programs do not have direct access to raw disk. File system are data structures that manages files-FAT table. File system contains the following modules:

* + - * Initialization
      * File system Operations
      * Memory Handler
      * Data management
      * Error Handler

File system provides facilities for accessing, organizing and further managing the data.

The application could be developed both by following SDLC life cycle and IIDM Methodology.

**1.2** **DOCUMENT CONVENTIONS**

Font Type: Arial

Headings:

Font size: 18 pt

Sub-Headings:

Font size: 14 pt

Body:

Font size: 10 pt

## Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **ACRONYM** | **MEANING** |
| FAT | File Allocation Table |
| SDLC | Software Development Life Cycle |
| IIDM | Iterative and Incremental Developing Model |

## References

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Document**  **Name** | **Type** | **Description** | **Version referenced**  **in this document** | **Owner**  **/Location** | **Hard**  **/Soft** | **Remarks** |
| File system Case Study | Business Requirements | This document refers to the  Problem statement,  Functional Specifications,  Technical Description of the  System. |  | Group D  iGATE | Soft |  |

## 

## Process for Requirements Capturing

The overall functionality of the File System was understood by studying the case study provided to the team, which had the functional and technical specifications to be implemented on the File system.

Several meetings were held to elicit the requirements, by asking questions and through discussions, in an attempt to coalesce the different views into a coherent set of requirements.

# Project Scope

## Scope

The Project Scope is to develop a file system which has the functionalities of creating files, displaying files, listing files, deleting files, modifying files, and keeping a log of all the errors in a file to provide an easy way for handling files from a user’s perspective.

## Out of Scope

|  |  |  |
| --- | --- | --- |
| Sr No. | Out of Scope | Category  (Functional/Technical/ Environmental/ Security/Users/Other) |
| 1. | Data corrupt error | Functional |
| 2. | Super Block | Technical |
| 3 | Inode | Technical |
| 4. | Multiple users can open same file. | Functional |
| 5. | File system should allow 10 files should open at time. | Functional |
| 6. | File system should close the file if timeout expires. | Functional |

# Project Description

## 3.1 Product Perspective

The File System product is a follow-on member of the File Handling hierarchy of operating systems.

## Product Function / Business Process Overview

* File system initialization- File System which contains a root file of 10 MB is created which stores files and allows various operations to be performed on those files. All the relevant data structures are initialized at the time of initialization.
* File System provide the following operations to be performed on files.
* Creation of files within file system.
* Display specific file.
* List the files stored in file system.
* Modify or overwrite files.
* Delete file from file system.
* Maintain an Error log file.

## Assumptions and dependencies

* Microsoft Visual Studio is present on the system.
* Memory will be available to create root file
* Memory will be available to create a new file.
* The File system is dependent on the disk availability, buffer memory, and operating system environment like Windows.

# Functional Requirements

## FUNCTIONALITY

|  |  |
| --- | --- |
| **ID** | **FUNCTIONAL REQUIREMENTS** |
| FR\_1 | The File System should be initialized for its first usage (i.e. if the file system is not present, then it needs to be created first).Initialization will involve assigning a size of 10 MB to the File System. |
| FR\_2 | The system should allow the user to create files and to store data in the files. |
| FR\_3 | If one or more file exists in the file system then the System should list all the file names with primary details. Details should include:   * Name of the file. * Timestamp of creation. * File size in kb. * Timestamp of file modification. * Block number (number of block where the file has been stored) * Permission on the file (read only permission/write permission). |
| FR\_4 | The file system should have an option to display the details of a specific file. The system should accept the file name and display the contents of that file and its primary details (mentioned in FR\_3). |
| FR\_5 | The file system should allow the user to delete a file present in the system (only if it is write mode).File System should accept the file name, check if the given file exists in the system and then delete the particular file after the user confirmation (if user is sure for delete operation).If the file is in read only mode,the system should not allow the user to delete it. |
| FR\_6 | The File System should have a provision to change the access permissions of the files. |
| FR\_7 | The File system should have a provision to edit the files .The following options should be included in edit. Append option: It allows appending the text at the end of the file. Overwrite option: It allows overwriting the previous content of file with newly entered content after user confirmation. |
| FR\_8 | To check consistency of the file system. |
| FR\_9 | File system should store maximum 100 files. File cannot be created or stored if there are already 100 files in the system. |
| FR\_10 | The designing of the system should be done in such a way that the later modifications or enhancement in functionalities will be easier to implement. |
| FR\_11 | The file system should operate in two modes :   * Read only mode. * Read/write mode.   File system mode should be decided at the time of initialization. |
| FR\_12 | The system should allow the following operations on each file :   * Open * Read * Write |
| FR\_13 | Every file in the system can be opened in the following mode:   * Read only. * Read/write |
| FR\_14 | If the file is kept idle for a particular time period i.e. no operations are performed /or no instructions are given on the file after opening it for that particular time, then the File should get close automatically after the timeout expires. |
| FR\_15 | The timestamp information should get stored.   * At the time of creation. * When the file is edited /modified. |
| FR\_16 | The Same memory (File system of 10 MB) is used for storing the file contents and the data structure's information which maintains the entire file system. |
| FR\_17 | If the data of the file needs to be overwritten or modified then a temporary file should be used to store the data which is to be written. The contents of the temporary file should get replaced with the original file contents after user confirmation. |
| FR\_18 | If there is any error encountered in the entire file system during user operations or due to system operations, then it should get caught and stored in the Error Log file |
| FR\_19 | The following errors should get captured by the file system:   * The File system mode is read only and the user tries to delete it or modify it. * Limited space available (i.e. if available space in file system is less than the new file's contents which is to be stored). * If file does not get saved due to any reason ( insufficient space error occurred due to any unexpected condition). * If file data gets corrupted.5. If file cannot be opened (due to some error encountered for e.g. file not found). |
| FR\_20 | File system should allow maximum 10 files to be opened at a time. |
| FR\_21 | Multiple users can open same file in read only mode. |

# Technical Requirements

|  |  |
| --- | --- |
| **ID** | **TECHNICAL REQUIREMENTS** |
| TR\_1 | The File System should have a root file of size 10 MB. |
| TR\_2 | The maximum number of files that can be created in the file system is 100 |
| TR\_3 | The maximum size of a file in the File system should be less than 10 MB |
| TR\_4 | Each file should have a data block to store the contents of the file. The data block size should be between 1KB-4KB. |
| TR\_5 | The File name length of each file in the File System should not be more than 255 characters. |
| TR\_6 | The File System should allow the user to open multiple files at a time. The maximum number of files that can be opened at a time is 10 |
| TR\_7 | The File System should have an Inode structure for each file in the system to store the basic information about the file |
| TR\_8 | The following information should be included in the Inode structure. |
| TR\_8.1 | File type: each file stored in the file system can be either a regular file or a directory file. |
| TR\_8.2 | File size: the size of the file should be stored in bytes. |
| TR\_8.3 | Timestamp of the file:this will include the time the file was created and the modification time, if any changes are made to the file. |
| TR\_8.4 | Access permission for a file:this will allow the user to open the file in either of the two modes -read only mode or read/write mode. |
| TR\_8.5 | Address of the first data block in the file where the contents of the file are stored. |
| TR\_9 | Every data block should have a unique sequence number to identify the file for which it has been created. |