**Algorithms and Data Structures Documentation**

**Overview**

This document provides detailed information about the algorithms and data structures implemented in the LockedMe.com application. Each algorithm is designed for optimal performance and maintainability.

**Data Structures**

***TreeSet<String> - Primary File Storage***

Purpose: Store file names in sorted order with efficient operations

* Memory efficient compared to maintaining separate sorted lists

***ArrayList<String> - File List Return***

Purpose: Convert TreeSet to List for indexed access and additional sorting

Characteristics:

* Random Access: O(1) indexed access for display
* Dynamic Sizing: Automatic capacity management

**Core Algorithms**

**File Addition Algorithm**

**Steps:**

1. Input validation and sanitization
2. Duplicate check using TreeSet.contains()
3. Physical file creation
4. Add to virtual file system

**Optimization Techniques:**

* Early termination on duplicate detection
* Atomic operation with rollback on failure
* Input validation

**File Deletion Algorithm**

**Safety Features:**

* Existence verification
* Case sensitivity

**File Search Algorithm**

**Algorithm Analysis:**

* Time Complexity: O(log n)
* Space Complexity: O(1)
* Search Method: Binary search in Red-Black tree

**File Listing Algorithm**

public List<String> getAllFilesSorted() {

List<String> sortedFiles = new ArrayList<>(virtualFileSystem);

Collections.sort(sortedFiles, String.CASE\_INSENSITIVE\_ORDER);

return sortedFiles;

}

Algorithm Analysis:

* Time Complexity: O(n log n)
* Space Complexity: O(n)

**Algorithm Design Decisions**

**TreeSet vs HashMap vs ArrayList**

***Chosen: TreeSet***

**Reasoning:**

* Automatic sorting
* Duplicate prevention
* Efficient search
* Memory efficient

**Case-Sensitive vs Case-Insensitive**

Implementation: Hybrid approach

* Internal: Case-insensitive
* Search: Case-sensitive
* Display: Case-insensitive

**Future Optimizations**

* Database Integration
* Caching
* Indexing
* Parallel Processing
* Lazy Loading