

LockedMe Documentation

Developer - Ananya Sharma

Contents:

1. *Specifications*
 - a. *Product Overview*
 - b. *Capabilities*
 - c. *Appearance and User Interaction*
2. *Sprints Overview*
 - a. *Sprint 1: Setup and Basic Functionality*
 - b. *Sprint 2: Business Operations and File Management*
 - c. *Sprint 3: Optimization and Testing*
3. *Java Concepts and Algorithms*
4. *Data Structures for Sorting and Searching*
5. *Application Flow and Flow Chart*

GitHub repository: <https://github.com/fr-ananya/Lockedme.com>

1. Specifications

Product Overview

The LockedMe File Management System is a Java-based console application designed to manage files within a specified directory. The application allows users to view, add, delete, and search for files. Additionally, it provides business operations for managing files in a user-friendly manner.

Capabilities

- **Display All Files:** Lists all files in the specified directory in ascending order.
- **Add a File:** Allows the user to add a new file to the directory.
- **Delete a File:** Enables the user to delete an existing file from the directory.
- **Search for a File:** Allows the user to search for a specific file in the directory.
- **Business Operations:** A submenu providing file management operations.

Appearance and User Interaction

- **Main Menu:** Displays the main options for the user to select.
- **File Operations Menu:** Provides options for adding, deleting, and searching files.
- **User Prompts:** Guides users through various operations with clear instructions.
- **Error Handling:** Catches and displays error messages for invalid inputs and exceptions.

2. Sprints Overview

The project will be completed in three sprints over the next 15 working days (3 weeks). Each sprint will last for 5 working days. Below is a detailed breakdown of each sprint:

Sprint 1: Setup and Basic Functionality

Duration: 5 days

Tasks:

1. **Project Setup:**
 - Initialize the project repository on GitHub.
 - Set up the development environment in Eclipse.
 - Create the basic project structure with appropriate packages.
 - Configure build tools and dependencies (if any).
2. **Implement Main Menu and Navigation:**
 - Create the main menu interface.
 - Provide navigation options for displaying files, accessing business operations, and exiting the application.
 - Ensure that user inputs are correctly read and processed.
3. **Implement List Files Functionality:**
 - Create a method to list all files in the specified directory.
 - Ensure files are listed in a sorted order (initially, sorting can be skipped; it will be handled in Sprint 3).
4. **Basic Error Handling:**
 - Implement basic error handling for invalid user inputs.
 - Ensure the application does not crash on invalid inputs.
 - Display user-friendly error messages.

Sprint 2: Business Operations and File Management

Duration: 5 days

Tasks:

1. **Implement Add, Delete, and Search File Functionalities:**
 - **Add File:** Create a method to add a new file to the directory.
 - **Delete File:** Create a method to delete an existing file from the directory.
 - **Search File:** Create a method to search for a specific file in the directory.

2. **Enhance Business Operations Menu:**

- Add options for add, delete, and search functionalities.
- Ensure smooth navigation between the business operations menu and the main menu.
- Provide clear prompts and feedback to the user.

3. **Advanced Error Handling and Validation:**

- Implement advanced error handling for file operations (e.g., handle file not found, file already exists, etc.).
- Validate user inputs to prevent invalid file names and operations.
- Ensure all exceptions are properly caught and handled with appropriate messages.

Sprint 3: Optimization and Testing

Duration: 5 days

Tasks:

1. **Implement Sorting Algorithms:**

- Implement a custom case-insensitive sorting algorithm (selection sort) for listing files.
- Ensure the list files method uses the custom sorting algorithm to display files in sorted order.

2. **Optimize Code for Performance:**

- Review and refactor the code for better performance and readability.
- Optimize file operations to handle large directories efficiently.
- Ensure the application follows best practices and coding standards.

3. **Testing and Debugging:**

- Conduct thorough testing of all functionalities.
- Identify and fix bugs and issues.
- Test edge cases and validate the robustness of error handling.
- Perform user acceptance testing to ensure the application meets the specified requirements.

4. **Final Documentation:**

- Prepare the final project documentation, including a user manual.
- Document the code with comments and explanations.
- Update the README file with project details, setup instructions, and usage guidelines.
- Ensure all documentation is clear, concise, and comprehensive.

3. Java Concepts Being Used

Core Concepts and Algorithms

- **Object-Oriented Programming (OOP):** Encapsulation, Inheritance, Polymorphism, and Abstraction.
- **Exception Handling:** Using try-catch blocks to handle input and file-related exceptions.
- **Collections Framework:** Utilizing sets, lists, and maps to manage files and records.
- **File I/O:** Reading from and writing to the file system.
- **Sorting Techniques:** Implementing a custom sorting algorithm (selection sort) for ordering files.
- **User Input Handling:** Using the `Scanner` class to read user inputs.

Data Structures and Techniques

- **Sets:** Used for storing unique books and authors.
- **Maps:** Used for maintaining borrow records.
- **Arrays:** Utilized for sorting files.
- **Search Algorithms:** Linear search for finding files.

4. Data Structures for Sorting and Searching

Sorting

- **Arrays:** Used for sorting file names in the `listFiles` method using a custom case-insensitive selection sort.

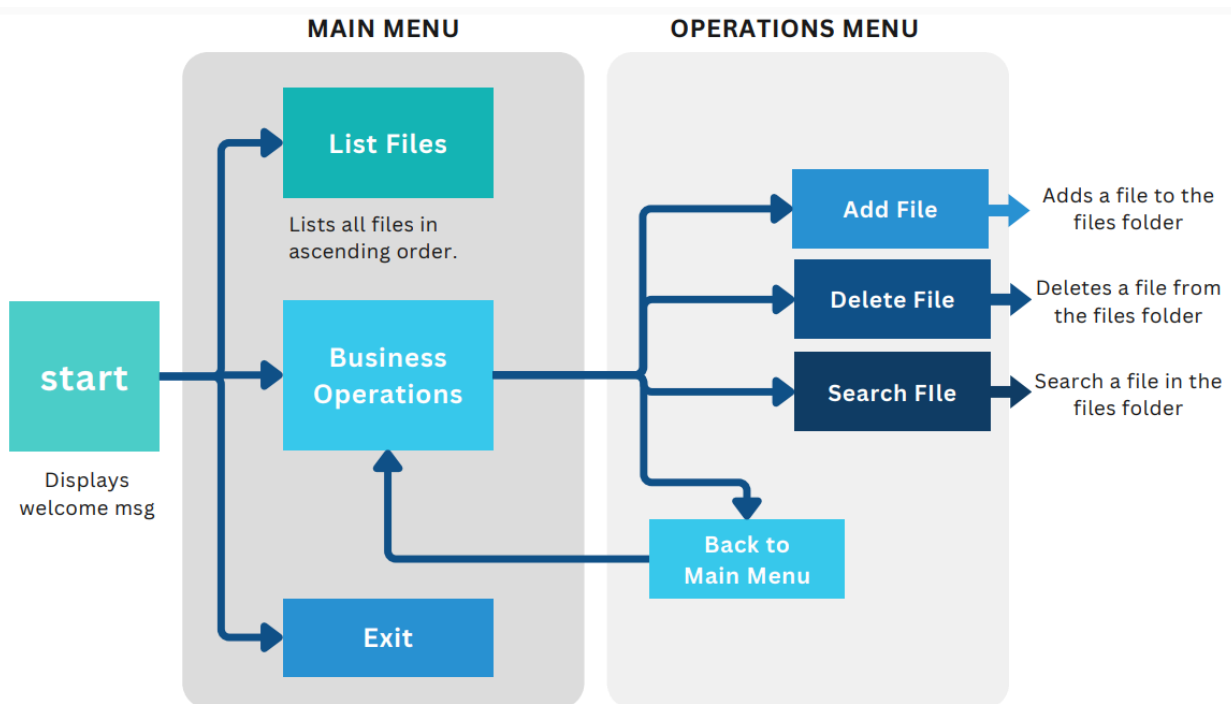
Searching

- **Linear Search:** Used in the `searchFile` method to find a specific file by name.

5. Application Flow and Flow Chart

Application Flow

1. **Start:** The application starts and displays the welcome message and main menu.
2. **Main Menu:**
 - **Option 1:** List all files (sorted).
 - **Option 2:** Navigate to the business operations menu.
 - **Option 3:** Exit the application.
3. **Business Operations Menu:**
 - **Option 1:** Add a file.
 - **Option 2:** Delete a file.
 - **Option 3:** Search for a file.
 - **Option 4:** Return to main menu.
4. **Exit:** The application exits.



Flowchart 1: a visual representation of the flow of the application.

```

Start
|
v
Display Welcome Message and Main Menu
|
+-----+
| 1. List Files          |----> List all files in sorted order
| 2. Business Operations |
| 3. Exit                |----> Exit Application
+-----+
|
v
Business Operations Menu
|
+-----+
| 1. Add a File          |----> Prompt user to enter file name and add file
| 2. Delete a File       |----> Prompt user to enter file name and delete file
| 3. Search for a File   |----> Prompt user to enter file name and search file
| 4. Return to Main Menu |----> Return to main menu
+-----+
|
v
Exit

```

Flow Chart2: A high level view of the application flow in text.