

Application Design and Planning Documentation

Table of contents

1. PROJECT OVERVIEW	
1.1. Purpose	3
1.2. Objectives	3
2. REQUIREMENTS ANALYSIS	
2.1. Functional Requirements	4
2.2. Non-Functional Requirements	4
3. USER FLOW DIAGRAM	
4. TECHNICAL SPECIFICATIONS	
5.1. Technical Stack	6
5.2. File Structure	6
5. IMPEMETATION PLAN	
5.1. Development Steps	7
6. TESTING PLAN	
6.1. Test Cases	8
7. USER GUIDE	
7.1. User Instructions.....	9
7.2. File Format	9
7.3. Know Limitations	9
8. MAINTENACE AND FUTURE ENHANCEMENTS	

1. Project Overview

1.1 Purpose

The purpose of this application is to provide a simple web-based tool that allows users to upload a .txt file containing customer details, search for specific customers, and display sorted customer information efficiently.

1.2 Objectives

- To read customer data from a .txt file.
- To display customer details in a user-friendly interface.
- To allow users to search for specific customers by name.
- To enable sorting of customer details alphabetically.

2. Requirements Analysis

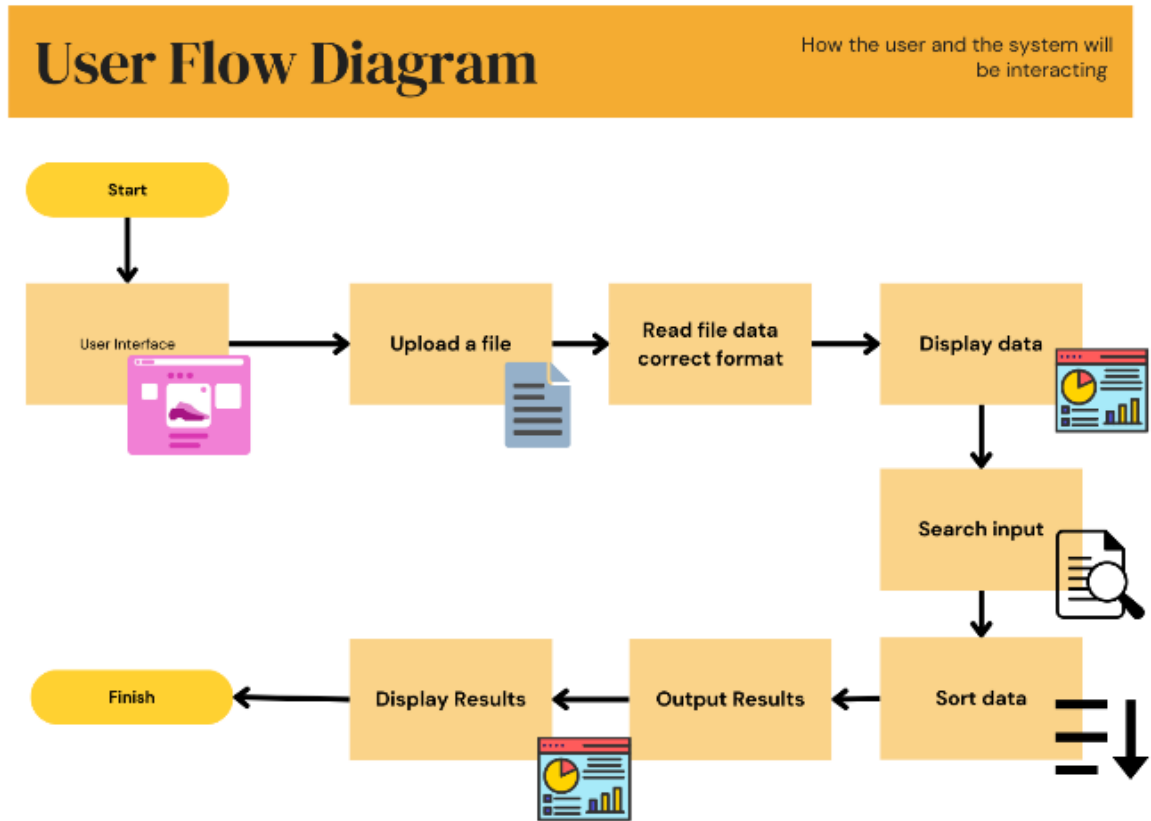
2.1 Functional Requirements

1. **File Upload:** Users can upload a .txt file containing customer data.
2. **Data Display:** The application displays the loaded customer data in a table format.
3. **Sorting:** Users can sort the displayed data alphabetically by customer names.
4. **Searching:** Users can input a search term to find specific customers.
5. **Output Results:** Display search results, including the index or position of the found item.

2.2 Non-Functional Requirements

- **Usability:** The interface should be intuitive and easy to navigate.
- **Performance:** The application should load and process files quickly, even with a large number of records.
- **Compatibility:** The application should work in all modern web browsers.

3. User Flow Diagram



4. Technical Specifications

4.1 Technology Stack

- **Frontend:** HTML, CSS, JavaScript
- **File Format:** .txt (Comma-separated values, e.g., "Name, Email, Phone")
- **Libraries:** None (pure JavaScript implementation)

4.2 File Structure

- **index.html:** The main HTML file that contains the structure of the application.
- **styles.css:** The stylesheet that contains all CSS rules for styling the application.
- **app.js:** The JavaScript file that contains the logic for file handling, sorting, searching, and displaying data.

5. Implementation Plan

5.1 Development Steps

1. Set up the HTML structure in index.html.
2. Implement file upload functionality in app.js.
3. Create a function to read and parse the uploaded file.
4. Display the loaded customer data in a table format.
5. Implement sorting functionality.
6. Create search functionality to filter customer data.
7. Style the application using styles.css.
8. Conduct testing to ensure all functionalities work as intended.
9. Document the application for user guidance.

6. Testing Plan

6.1 Test Cases

1. File Upload Test:

- **Input:** Valid .txt file.
- **Expected Result:** Data loads successfully.

2. Sorting Test:

- **Input:** Customer data in random order.
- **Expected Result:** Data displays in alphabetical order upon sorting.

3. Search Test:

- **Input:** A customer name.
- **Expected Result:** Displays matching customer(s) and their position in the list.

4. Error Handling Test:

- **Input:** Empty file or unsupported format.
- **Expected Result:** Appropriate error message displayed.

7. Documentation and User Guide

7.1 User Instructions

- **How to Use:**

1. Click the "Upload File" button to select your customer .txt file.
2. After the file uploads, the customer data will be displayed in the table.
3. Use the search box to find a specific customer by name.
4. Click the "Sort" button to sort the customers alphabetically.

7.2 File Format

- Ensure your .txt file is formatted as follows:

Name, Email, Phone

John Doe, john@example.com, 1234567890

Jane Smith, jane@example.com, 0987654321

7.3 Known Limitations

- The application currently supports only .txt files.
- The search function matches exact names; it may not support partial matching.

8. Maintenance and Future Enhancements

- Consider adding support for CSV file formats.
- Explore features like exporting search results and improving the UI/UX for mobile responsiveness.