Java III AT3 – Product Specification Design Document

L. E. Liston

2020

Contents

[1 Introduction 1](#_Toc43633329)

[2 Description 1](#_Toc43633330)

[3 Implementation Plan 1](#_Toc43633331)

[4 Source Control 1](#_Toc43633332)

[5 Test Plan 1](#_Toc43633333)

[5.1 Testing Tools 1](#_Toc43633334)

[5.2 Testing Environment 1](#_Toc43633335)

[5.3 Test Cases 2](#_Toc43633336)

[6 Functional Requirements 4](#_Toc43633337)

[7 Non Functional Requirements 4](#_Toc43633338)

[8 Quality Assurance Practices 4](#_Toc43633339)

# Introduction

This project will be created to showcase a range of different features. It will be developed in Java using the Netbeans IDE.

# Description

The program will use linked lists in order to scroll through multiple options in a set of data.

It will use hashing techniques on multiple strings and compare the two.

It will sort a list of string entries using the Mergesort as opposed to any other for its simplicity and relative optimization.

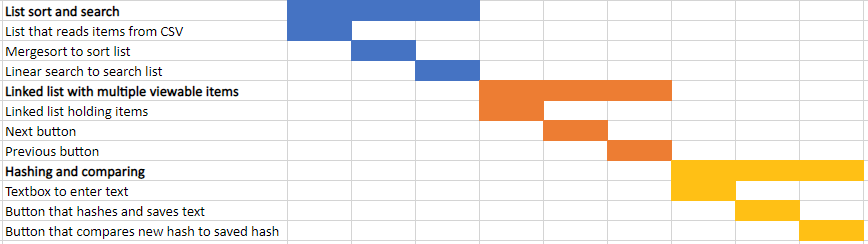
The linear search method will be used over the more complex binary search due to the list being relatively small. It also means that the list can be searched without the list being sorted.

The entries will be read by the third party OpenCSV reader which can be found here: <http://opencsv.sourceforge.net/>

The entire program will be made as a GUI application supported by Netbeans.

# Implementation Plan

Below is an image of the implementation plan for this project.



# Source Control

All project files will be uploaded Github at this link: <https://github.com/M227545/JavaIIIAT3_Project>

# Test Plan

## Testing Tools

Automated testing tools used to generate test information without the input of a developer.

Built in debugging tools manually accessed by developers to fix bugs or errors.

## Testing Environment

Windows 10 computer.

Netbeans IDE.

## Test Cases

|  |  |  |
| --- | --- | --- |
| Test case | Expected outcome | Screenshot |
| Click sort button | The list is sorted alphabetically |  |
| Type “May” then click search button | The item “May” in the list is highlighted |  |
| Type “Month” then click search button | No new item is highlighted |  |
| Click next button | The next item in the linked list is displayed |  |

|  |  |  |
| --- | --- | --- |
| Click previous button | The previous item in the linked list is displayed |  |
| Click next button at end of list | The item displayed is not changed after the last item |  |
| Click previous button at start of list | The item displayed is not changed after the first item |  |
| Type “hello” into textbox then click hash | The text “hello” is hashed and saved | The textbox is cleared automatically after pressing the hash button |

|  |  |  |
| --- | --- | --- |
| Type “hello” into textbox then click compare | A success message is displayed |  |
| Type “hi” into textbox then click compare | A failure message is displayed |  |

# Functional Requirements

The functional requirements are to use dynamic data structures, hashing, sorting, searching, a third party library and a GUI for the program.

# Non Functional Requirements

The non functional requirements are to quickly sort data with an efficient sorting method. The search must also be fast to not keep the user waiting for the program.

# Quality Assurance Practices

The entire program will be tested thoroughly using a test table, ensuring that all pathways are tested. The program will also be debugged using the built in Netbeans IDE debugging functionality.