**Software Requirements Specification**

***FileSearch App***



CPSC 362

Professor: Dr. Lidia Morrison

Department of Computer Science

California State University, Fullerton

Fall, 2019

Grae Lessley Kevin Chen

Eric Tran Lambert Liu Josh Ungheanu

**Table of Contents**

[**1. Introduction** 4](#_Toc27005663)

[**1.1 Purpose** 4](#_Toc27005664)

[**1.2 Scope of the Problem** 4](#_Toc27005665)

[**1.3 Intended Audience** 4](#_Toc27005666)

[**2. Overall Description** 5](#_Toc27005667)

[**2.2 Product Functions** 5](#_Toc27005668)

[**2.3 Operating Environment** 6](#_Toc27005669)

[**2.4 Similar System Information** 6](#_Toc27005670)

[**2.5 User Characteristics** 9](#_Toc27005671)

[**2.6 Design and Implementation Constraints** 9](#_Toc27005672)

[**2.7 Assumptions** 9](#_Toc27005673)

[**3.0 Functional Requirements** 10](#_Toc27005674)

[3.1 The app shall display a widget 10](#_Toc27005675)

[3.2 The app shall display a search bar 10](#_Toc27005676)

[3.3 The app shall let the user type in a keyword 10](#_Toc27005677)

[3.4 The app shall let the user select from the dropdown list 10](#_Toc27005678)

[3.5 The app shall display the search results 11](#_Toc27005679)

[3.6 The app shall allow the user to type in another search 11](#_Toc27005680)

[3.7 The app shall display a second list of results 11](#_Toc27005681)

[3.8 The app shall allow the user to clear the list 12](#_Toc27005682)

[3.9 The app shall display in the GUI the content of the file 12](#_Toc27005683)

[3.10 The app shall highlight the keyword in the GUI 12](#_Toc27005684)

[3.11 The app shall allow the user to select another item from the list 12](#_Toc27005685)

[3.12 The app shall allow the user to close out of the app 13](#_Toc27005686)

[**4.0 Non-Functional Requirements** 13](#_Toc27005687)

[**5.0 Interface Requirement** 13](#_Toc27005688)

[5.1 GUI 13](#_Toc27005689)

[5.2 Hardware Interfaces 15](#_Toc27005690)

[5.3 Software Interfaces 15](#_Toc27005691)

[**6.0 SWOT Analysis for FileSearch** 15](#_Toc27005692)

[**7.0 UML Diagrams** 17](#_Toc27005693)

[7.1. Use Case Diagram 17](#_Toc27005694)

[**APPENDIX A: USER OPERATIONS MANUAL** 18](#_Toc27005695)

[**APPENDIX B: References and Tools Used** 21](#_Toc27005696)

**Table of Figures**

***Figure 1 – Search UI Interface……………………………………………………………………………..… 8***

***Figure 2 – Search Results GUI……………………………………………………………………………..… 8***

***Figure 3 GUI Diagram……………………………………………………………………………..………….14***

***Figure 4 Use Case Diagram………………………………………………………………………..………….18***

***Figure 5 Screenshot of initial screen……………………………………………………………..…………. 19***

***Figure 6 Screenshot of search….………….……………………………………………………..…………. 20***

***Figure 7 Screenshot of* Drop-down list *….………….……………………………………………..…………. 21***

***Figure 8 Screenshot of content….………….……………………………………………………..…………. 21***

# 1. Introduction

## **1.1 Purpose**

The following document offers a description of the software requirements for our 362-group project. The document will explain the general purpose and reasoning behind our windows application, FileSearch, including the software requirements and overall goals to be accomplished.

## **1.2 Scope of the Problem**

We all use computers as a way to store and process data and the world has become less dependent on physical storage, many of us not realizing that we store everything from homework assignments to personal passwords to a computer. This is where FileSearch comes in by introducing an easy and user-friendly alternative to search for keywords in a specific file.

Like its competitors, FileSearch will provide the minimum features necessary in order to provide an enjoyable and comprehensive file tracking experience. FileSearch provides the user the ability to search a keyword through all directories and once all files are found, the users will be able to select which file they would like to load and preview it with our GUI.

## **1.3 Intended Audience**

The Intended Audience of this document includes Professor Lidia Morrison, and the members of our group to verify the functionality of the software. Other users include all students enrolled in CPSC 362 for (Fall 2019) at California State University, Fullerton. The application is intended for individuals in need of a finding a specific keyword within their directories of all file types.

# 2. Overall Description

**2.1 User Objectives**

FileSearch is a windows application that focuses on providing a medium by which a user can search a specific keyword at which all files within all directories are pulled into a scroll bar option list. From here the user will select one of the following options and the app will grab all the text within the file and display it in the GUI. The searched keyword will be highlighted. Figure 1 shows the FileSearch interface.

## **2.2 Product Functions**

FileSearch is a windows application that is centered towards any users in general. With the app, users can search for any file, whether its work related or for personal use that is saved on any drive including external drives. Users can select from any directory and focus on files directly from one drive. Users can also reset the search by typing in another keyword in the search field section. This will start the search process all over again.

Below are the basic functions of FileSearch:

* Users can search for any keywords within all drives.
* The app will only detect keywords from certain file types.
  + .txt
  + .xlsx
  + .pptx
  + .docx
* The app will allow users to scroll through a list of all files that contain the keyword being searched.
* All content from the file will be transferred to the GUI and the keyword will be highlighted.

## **2.3 Operating Environment**

FileSearch is a windows application that can be deployed using an executable file. It was

generated by pyinstaller, which freezes (packages) Python applications into stand-

alone executables, under windows, GNU/Linux, Mac OS X, and other operating systems.

The Target OS (Operating System) for this project is Windows but we were able to run the search function on a mac. That being said, executables allow for an easy deployment to all windows platforms including windows 7, 8, and 10. Deployment on non-windows platforms is scheduled for future releases of FileSearch.

## **2.4 Similar System Information**

There are over 100 apps for file searching applications on the window market. Topping the list are: Wise JetSearch, Everything, Quick Search, SearchMyFiles, and File Seek (Source: https://www.lifewire.com/11-free-file-search-tools-1356644).

A screenshot of a social media post

Description automatically generated

***Figure 1 – Search UI Interface***

A screenshot of a cell phone

Description automatically generated

***Figure 2 – Search Results GUI***

## **2.5 User Characteristics**

This app is for users interested in searching for a keyword within a file. The app has no age restriction. It can be used by everyone. There are no gender restrictions with the app usage. No special skill is required in operating the app.

## **2.6 Design and Implementation Constraints**

The app is compatible with only Windows devices. This includes devices running Windows 7 and up. Devices include Windows phones, tablets, laptops, and desktops. Users with other devices may be required to install a virtual Windows OS for FileSearch to perform adequately. FileSearch is a Windows app and is not expected to run on a mobile device. Input devices such as mouse and keyboard are required to use the app since it will be built solely for Windows devices. External speakers are also not required in using the app. The app is not a game and does not provide any form of entertainment to users.

## **2.7 Assumptions**

## 

This SRS assumes that:

* Users have a prior knowledge in operating a windows device
* User’s Window’s device runs on the latest operating system
* Users have a mouse and keyboard
* User understands English
* Users are able to read and understand information displayed on widgets
* Users understands directories and how paths are defined

# 3.0 Functional Requirements

## 3.1 The app shall display a widget

**Description**

On clicking the app icon, a windows screen is displayed.

Pre-condition – User has downloaded and installed FileSearch

Post-condition – The main UI page is shown

## 3.2 The app shall display a search bar

**Description**

A widget page is displayed, and the user is prompted to enter a keyword.

**Pre-condition** – User has a valid Windows OS

**Post-condition** - The search will initiate if keyword is valid, else the app will close.

## 3.3 The app shall let the user type in a keyword

**Description**

The app shall let the user enter a keyword in the search field

**Precondition** – User can select the search field with the mouse

* **Post-condition** – The user is required to enter the details of the keyword that is associated with any file type within our scope page. This include:
  + .txt
  + .xlsx
  + .pptx
  + .docx

## 3.4 The app shall let the user select from the dropdown list

**Description**

After entering the keyword, the user can select one of the options by selecting the dropdown field. The field contains an arrow pointing down.

**Pre-condition –** User selected drop down arrow field

**Post-condition –** Option selected is registered and GUI begins to gather the content from the file.

## 3.5 The app shall display the search results

**Description**

The FileSearch GUI contains a retrieve box that displays the content of the file that was selected from the drop-down search results. The screen contains all of the content that is in the file. Excel files we formatted to mimic cells so the data will match with what you see in the actual file. Easier for users to find the highlighted keyword.

**Pre-condition –** User was able to find the file path from the dropdown list. Making sure necessary drives were installed.

**Post-condition –** User can find the specific keyword they were searching for.

## 3.6 The app shall allow the user to type in another search

**Description**

The User can easily type in another search keyword in the search bar. The application will clear the cache and restart the process again. Searching through all drives, including external if available.

**Pre-condition –** User finds the search bar.

**Post-condition –** User is given a new result.

## 3.7 The app shall display a second list of results

**Description**

After entering the keyword, the user can select one of the options by selecting the dropdown field. The field contains an arrow pointing down.

**Pre-condition –** User selected drop down arrow field

**Post-condition –** Option selected is registered and GUI begins to gather the content from the file.

## 3.8 The app shall allow the user to clear the list

**Description**

When the user enters a different keyword, the list is cleared, and cache has been reset.

**Pre-condition –** a keyword is required for the list to be cleared.

**Post-condition –** User experiences a new file search

## 3.9 The app shall display in the GUI the content of the file

**Description**

The content of the file selected will display in the GUI as a separate section.

**Pre-condition –** User selects the path in which the file is in.

**Post-**condition – User will be able to see the highlighted keyword within the content of the file.

## 3.10 The app shall highlight the keyword in the GUI

**Description**

The app will highlight and display the keyword within the GUI.

**Pre-condition –** User selects a file within the drop-down list generated by the search

**Post-condition –** Content and highlighted word continues to display until new keyword is entered.

## 3.11 The app shall allow the user to select another item from the list

**Description**

The user can select another item from the drop-down list and view other files with the same keyword.

**Pre-condition –** User selects the field with the drop-down arrow.

**Post-condition –** A new view of the content will display in the GUI along with the highlighted keyword.

## 3.12 The app shall allow the user to close out of the app

**Description**

The user can close the app by hitting the “x” button from the top menu.

**Pre-condition –** User selected the “x” button from the top menu.

**Post-condition –** The app closes out.

# 4.0 Non-Functional Requirements

**Security:**

* The system ***will*** require the user have permission/access to all files within the computer.
* Any encrypted files will not be accessed.

**Usability:**

* After starting the application, the software ***will*** finish searching within 20-40 seconds.
* When switching between searches, the system ***will*** load the next search within 20-40 seconds.

# 5.0 Interface Requirement

## 5.1 GUI

When a user selects the FileSearch exe on their device, a screen is displayed. What follows is a widget screen waiting for the user to input a keyword in the search field. After the user does so, the search function will begin search all files within all available drives. Using a multithread process, we can save 2-4 times the speed. Since we have multiple threads searching through the drives. The user can then select from the drop-down option one of the files that contains the keyword. The drop-down will display the path of the file as well so user can easily follow it and find the actual file if needed. The GUI will then pull all the content from the file and display in in a box. Any keyword found in the content will be highlighted red. You can then search for another keyword and the list will clear and start the process over again.

Search Dropdown

[A screenshot of a social media post

Description automatically generated](https://cdn.discordapp.com/attachments/617054960168992772/654604187916828687/unknown.png) [A screenshot of a social media post

Description automatically generated](https://cdn.discordapp.com/attachments/617055149915111456/652734392007000084/Step_1.png)

**Figure 4 GUI Diagram**

## 5.2 Hardware Interfaces

This application is expected to function optimally on a Windows device, including tablets and phones. The app requires optionally that devices should have working mouse and keyboard, as the software usage is accompanied by movement and typing. ***Please note***: This requirement does not hinder the basic functionalities of FileSearch.

Before downloading and installing the app, a minimum of 20Mb internal storage space is required to be available. The app is expected to work best on devices with CPU speed of 2GHz and 4GB of RAM.

## 5.3 Software Interfaces

The Software interface of this app is already stated in the “Operating Environment” section above.

# 6.0 SWOT Analysis for FileSearch

**Strengths:**

S1: Since the scope of this project is not so wide, the expected functionality of the app can realize before the product’s deadline. This means that all the functional requirements will be met.

S2: The software tools and techniques required for developing this app is readily available and is within our reach.

S3: Since this project is a course work in the 362 class, it requires little or no form of financial support. Therefore, we can say that this project is economically feasible.

**Weakness:**

W1: Although this group consists of students with a good background in software design and also good programming skills, this is the first time we are building a windows application using python. This is actually a weakness because more time is spent learning how to use the software tools instead of producing the product. This may actually affect the project’s schedule.

W2: Since this is the first version of the app, it lacks certain functionalities compared to other budget assistant apps in the market. But subsequent versions will have more functionalities.

**Opportunities:**

O1: Almost everybody creates files and save data on their computer on a daily basis. Most individuals find it difficult to keep track of their information. This application therefore fulfils the interested user’s need for searching for forgotten information that might be lost in a path that can be frustrating to find.

O2: This project creates a leverage for users to find information that had been lost overtime. Some people are not well organized so searching for a file can take time and time is money.

**Threats:**

T1: Since this is a course work, other groups in the class are also developing a similar application. This is a threat because our team might be more focused on producing the best application by adding too many extra functions rather than producing the main product’s functionalities.

T2: There are also a lot of budget assistant applications already in the market and these products have a high recommendations and user ratings. This is a threat because it will be difficult for our new app to be able to break the market’s entry barrier.

# 7.0 UML Diagrams

## 7.1. Use Case Diagram

While on the main application screen, the user is presented with 4 options which provide functionality to the application. When the app is open, the user is presented with a screen that allows for the entering of a keyword that will then allow the application to start searching through all drives for files that contain that keyword. The app allows users to select from a list of files that contain the keyword, the nice thing is that the file contains the full path in which it is located.

A close up of a map

Description automatically generated

**Figure 4 Use Case diagram for File Search**

# **APPENDIX A: USER OPERATIONS MANUAL**

This section of this document provides an understanding of how to use the FileSearch app.

1. **Initial Screen**

On loading the app from the device’s menu, the File Search widget screen is displayed.

[A screenshot of a social media post

Description automatically generated](https://cdn.discordapp.com/attachments/617055149915111456/652735549194633247/Excel_result.png)

**Figure 5 Screenshot of the initial screen**

1. **Search**

Enter keyword and hit search. App will take 20-40 seconds to search through all drives and display the files that contain the searched keywors.

[A screenshot of a cell phone

Description automatically generated](https://cdn.discordapp.com/attachments/617054960168992772/654603930059276298/unknown.png)

**Figure 6 Screenshot of the search**

1. **Select File**

From the drop-down list, select the file that contains the keyword. Drop-down list will

contain the path of the file and which drive it is in.

[A screenshot of a social media post

Description automatically generated](https://cdn.discordapp.com/attachments/617055149915111456/652734392007000084/Step_1.png)

**Figure 7 Screenshot of the Drop-down list**

1. **Display Results**

Content of file will display in the retrieve box and all searched keywords will be highlighted red. Currently we do not support modifying the actual file.

[A screenshot of a social media post

Description automatically generated](https://cdn.discordapp.com/attachments/617054960168992772/654604388069015553/unknown.png) [A screenshot of a cell phone

Description automatically generated](https://cdn.discordapp.com/attachments/617054960168992772/654604286357012480/unknown.png)

**Figure 8 Screenshot of the Content**

# **APPENDIX B: References and Tools Used**

**References**

Lecture Notes and Slides by Lidia Morrison

https://www.pyinstaller.org

https://pythonspot.com/category/pyqt5/

https://www.lifewire.com/11-free-file-search-tools-1356644

**Tools Used**

The following tools were used for developing this SRS

* PYQT5 Widget
* Draw.io