File Stalker

Collaborative Team Project

Stage II – Requirements Modeling and Analysis

CSC260

Dr. Pulimood

16 March 2012

Team C.H.I.M.P

Alexa Cain

Danny DeChiara

Derek Duchesne

Tim Lee

Glen Oakley

Steve Schwarcz

**Problem Statement**

Users may have difficulty tracking changes to files on their hard disk. Program installations, background programs, and other processes may edit files without the user’s knowledge or consent. In the event of an undesired event or a system crash, the user may want to understand the changes in files leading up to the crash.

**System Objective**

Overall, File Stalker will help the user track changes to files.

Specifically, the system should be able to:

* Monitor the creation, deletion, and modification of files
* Monitor changes in files’ directories
* Maintain a running list of every change and when it was made
* Allow for querying of the stored data, including filtering by name, directory, ranges of dates, and various other filters

To aid in viewing the logs, the system will have the ability to filter results. For example, it may be unnecessary to monitor constantly changing operating system files.

**Desired End Product**

File Stalker will run in the background of a Unix system and will be able to deliver a log of all file modifications to the user. The user should be able to tailor the results for maximum benefit. It will be written in C++ with completely original code. It will integrate several classes with aggregation and inheritance relationships and implement polymorphism.

**System Boundary (see “Diagrams” page for a graphical representation)**

Modules:

* Obtain a list of changes (creation, deletion, and modification) to the system
* Sort data according to date, directory, file name, and the type of system change
* Record user-input for search tags (date, directory, file name, and type of change)
* Parse change log based on user criteria
* Output specified system changes

Not implemented:

* Storing older versions of files or directories
* Reverting changes made to the system
* Allowing the user to modify files from the File Stalker interface

**Networking/Shared Access Requirements**

File Stalker works locally, only monitoring changes on the machine on which it is

installed. It does not need Internet or network access. If a user wishes to back up logs

to another location or send them to someone else, he or she can manually access the log

files.

**Performance**

Our method of storing directory locations will be optimized for efficiency. We will represent directories in the system using trees, which will allow us to use graph traversal to navigate to the correct directories.

**Security**

The user will not be able to make any changes to files from File Stalker itself, as the program will only be responsible for generating and outputting a list of these changes. File Stalker will, however, tell the user when they have requested a file or directory that does not exist or if they have entered invalid search parameters. In either case, the user will be prompted again for correct input.

**Backup and Recovery**

File Stalker will be constantly monitoring system files and regularly saving over its logs so that if the program shuts down unexpectedly the system changes that had been recorded up to that point will not be lost. The user will then have to manually restart the program to continue monitoring the system.

**Legal Issues**

It should be made very clear to the end user what this program does and what it doesn’t

do. File Stalker is used to track file modifications, but it does not actually back up any

files. Users should be informed that File Stalker will perform to the best of its ability,

with no guarantees of its performance or liability in the event of failure. All of this should

be clearly stated in the End User License Agreement. Care must be taken to avoid File

Stalker from being easily adapted for malicious purposes.

**Possible Applications**

File Stalker has various personal and commercial applications. Users may want to

monitor activity on their computer, to see if there was been (physical) unauthorized

access or see what file changes led to an unexpected event. They may be able to check

for the presence of malware by monitoring logs, and seeing what changes occur without

their action or permission. There are applications specific to commercial or enterprise

environments. An administrator can see what an employee has been working on during

company time. With this information, an analysis can be conducted of how much effort is

used on a given task, giving the company insight on all of its employees’ time and effort

expenditures.

**Use Case Descriptions (see “Diagrams” page for graphical representation)**

**Use Case: Log Searching**

**Primary actor:** End user.

**Goal in context:** Searching for all file modifications from specified directory.

**Preconditions:** System must be configured, and have the ability to access the directory. The search tokens must be valid.

**Trigger:** User wants to check if a file was inadvertently deleted, background processes are making file changes without permission, or file changes led to a system crash.

**Scenario:**

1. User opens File Stalker frontend interface.
2. User enters the command to search log files.
3. User inputs search tags and/or specifies filters.

**Exceptions:**

1. Tag search returns no results.
2. Log file not found.

**Priority:** High priority, but not to interfere with normal logging.

**When available:** Available after File Stalker has begun logging specified directory.

**Frequency of use:** High frequency.

**Channel to actor:** Text-based command line interface.

**Secondary actors:** N/A

**Use Case: Changing Logged Directories**

**Primary actor:** End user.

**Goal in context:** Changing the directories to be logged by File Stalker.

**Preconditions:** The user has already specified valid directories to log.

**Trigger:** User wants to change which directories will be monitored by the program.

**Scenario:**

1. User opens File Stalker frontend interface.
2. User enters the command to change the currently monitored directory.

**Exceptions:**

1. An invalid directory was input.

**Priority:** High priority, but not to interfere with normal logging.

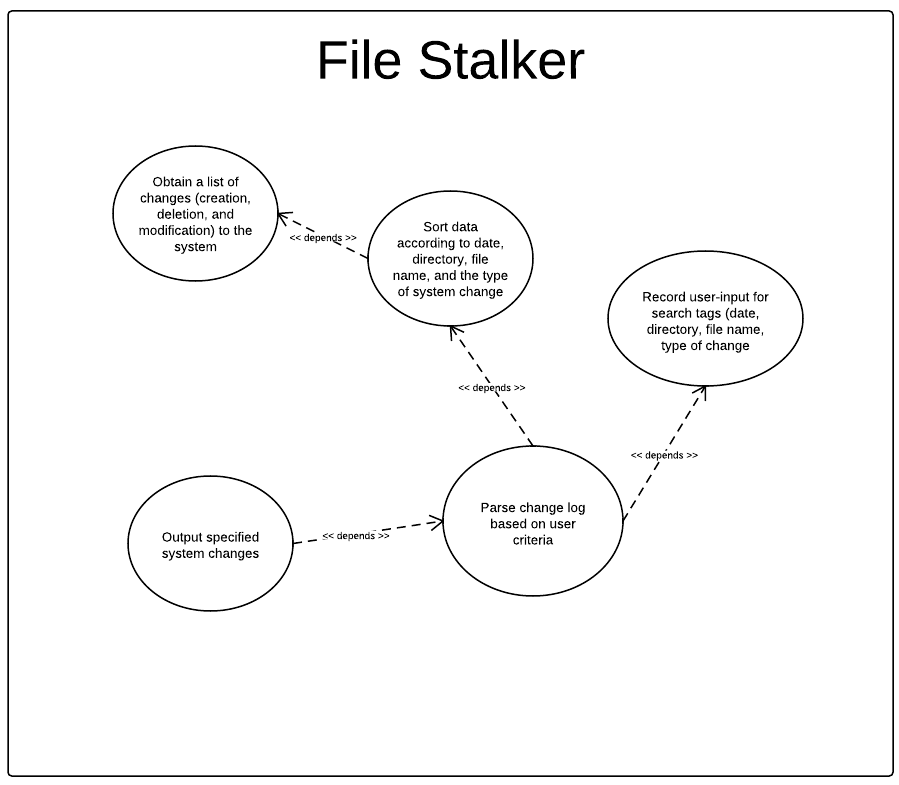
**When available:** Available at all times.

**Frequency of use:** Low frequency (File Stalker’s default monitored directory will be the user’s home directory).

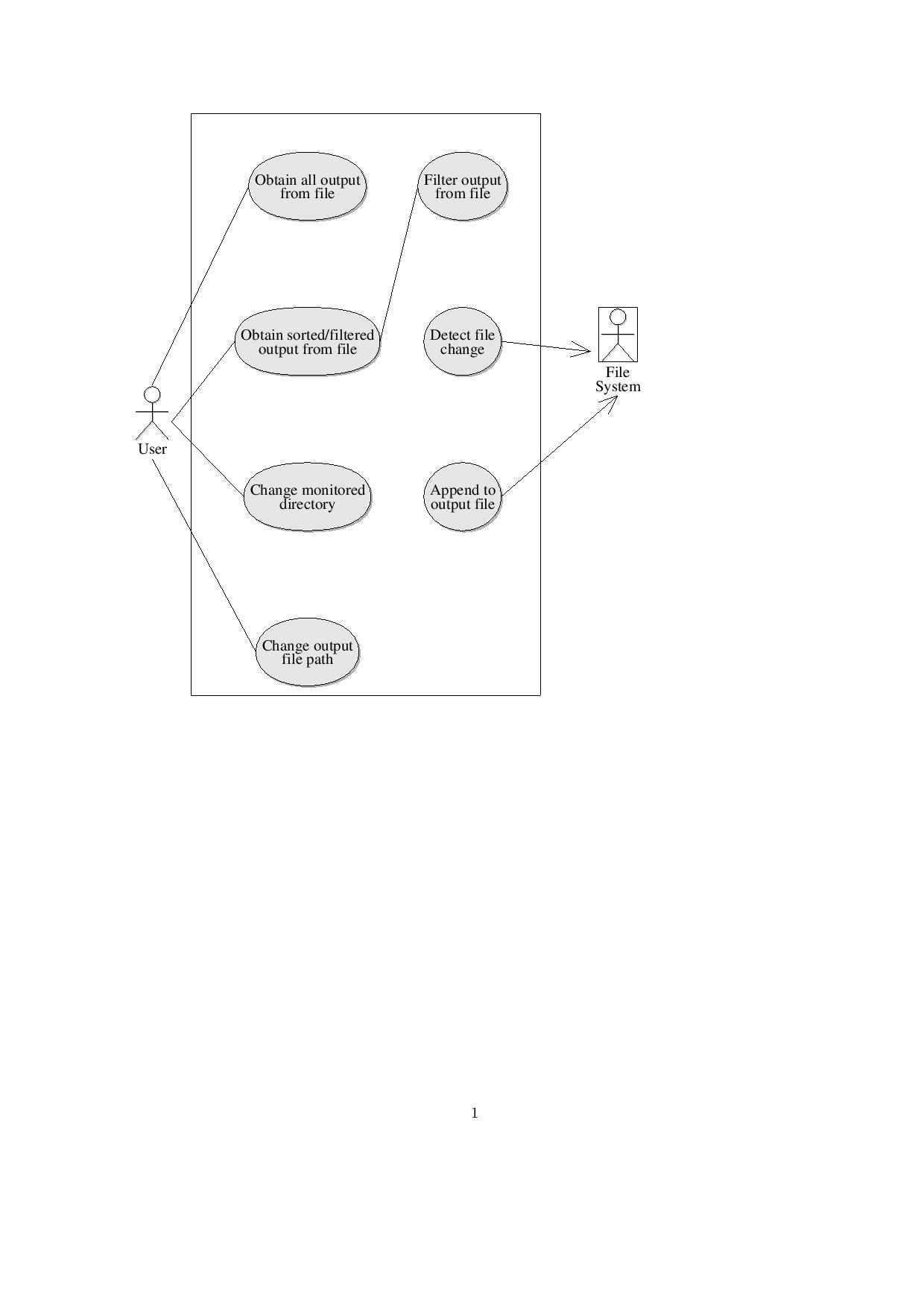
**Channel to actor:** Text-based command line interface.

**Secondary actors:** N/A

**Diagrams**

System Boundary:

Use Case:



Analysis Class:

