**SOFTWARE REQUIREMENTS SPECIFICATION**

Ahmed Mawari

Hasan Alameh,

Adham Abdalla,

Tanis Daniels-Wanamaker

[**1.0 Introduction**](#_heading=h.temg9zvnut5p) **4**

[1.1 Goals and objectives](#_heading=h.3z2jaeyjmy56) 4

[1.2 Statement of scope](#_heading=h.syjcslv0auoc) 4

[1.3 Software context](#_heading=h.2k66ctalag8r) 4

[1.4 Major constraints](#_heading=h.1hgjxgjjbo17) 4

[**2.0 Usage scenario**](#_heading=h.7u68xy2h4nd5) **5**

[2.1 User profiles](#_heading=h.l0ixh54mf3jt) 5

[2.2 User stories](#_heading=h.oglhitzcpact) 5

[2.3 Special usage considerations](#_heading=h.g3kwayyy33ka) 5

[**4.0 Functional Model and Description**](#_heading=h.gitqsuhicl1l) **6**

[4.1 Description for Function](#_heading=h.4jw9i771m9jg) 6

[4.1.1 First Use Case](#_heading=h.dpvig4tcttqp) 6

[4.1.2 Second Use Case](#_heading=h.gfi349hs76yu) 6

[4.1.3 Third Use Case](#_heading=h.jtwtosf0jrpy) 7

[4.1.4 Fourth Use Case](#_heading=h.tu6l1cnn1fj8) 8

[4.3 Use Case Diagrams](#_heading=h.yjbuef1ywckt) 9

[**8.0 Appendices**](#_heading=h.v09qhyo9hu23) **11**

[8.1 System traceability matrix](#_heading=h.xfj95f9t7a3) 11

# 1.0 Introduction

## 1.1 Goals and objectives

Our objective is to further develop the program’s flexibility so that more cities, counties, townships, and villages’ Financial Audits formats are properly flagged, extracted, and organized to minimize both user effort and human error.

## 1.2 Statement of scope

**Major Inputs:** -Scannable PDF files containing Financial Audit statements

**Processing Functionality:**

1. Program will read in a PDF file and save the Local Government name and year of submission.
2. Unneeded sections/pages will be skipped
3. Program will find sections within the user’s interest and flag the corresponding values

**Major Outputs:** -All flagged/extracted numbers will be appended to a .csv file under the appropriate column.

## 1.3 Software context

The software will allow the Michigan Department of Treasury to parse the different challenges communities are facing by tracing their financial funding/spending. The user will be able to use the extracted data for calculations and analysis.

## 1.4 Major constraints

-One major constraint is that there are 1000+ local governments, each with possibly different formats and may use different titles for the same data fields. It can be challenging to make the program flexible enough to properly flag and extract any imported format.

# 2.0 Usage scenario

**This section provides a usage scenario for the software. It organized information collected during requirements elicitation into use-cases.**

## 2.1 User profiles

**The profiles of all user categories are described here.**

User Profile: Data Analyst

A data analyst uses extracted data to produce specific statistics and key correlations in pursuance of making more informed decisions. The data analyst may use data from submitted financial audit forms in place of, or as a comparator to, the standardized and error-prone F-65 forms.

Data analysts are knowledgeable about examining, cleansing, and manipulating data, typically in a spreadsheet-like format. They may have little to no knowledge of the inner workings of data mining programs.

## 2.2 User stories

**All the user stories defining the use-cases for the software are presented using the user’s own words.**

-As a data analyst, I want to be able to select a PDF for the program to process so that the program can extract my desired data into an organized .csv file.

-As a data analyst, I want to be able to load different financial audit formats so that the program can process the non-standard audits into something more standardized.

-As a data analyst, I want to be able to track the extraction process so that I can see where the program has an error if one occurs.

## 2.3 Special usage considerations

**Special requirements associated with the use of the software are presented.**

-The file selected to be processed by the program must be in a readable, non-scanned PDF file.

-The selected file must be a local file on the Analyst's hard drive.

# 4.0 Functional Model and Description

## 4.1 Description for Function

### 4.1.1 First Use Case

### Use case name: File Upload

### Actors: User

### Preconditions: The user must already have the files (PDF) on the computer they are initiating the program on

### Triggers: The function will be fully utilized by the system when the user searches for the file through its file path and clicks upload

### Scenario Description:

1. The user initiates the program by clicking on the program’s application icon on their computer
2. The user clicks on the file upload option within the program
3. The user searches for the file they want to upload and find its file path
4. The user selects the file they want to upload and click upload

### Post Conditions: The system can now extract data from the file uploaded

### Exceptions: If the user uploads a different file format besides a PDF, the program should respond by not accepting the document

### 4.1.2 Second Use Case

### Use case name: Extract File Data

### Actors: System

### Preconditions: File upload

### Triggers: The function will be fully utilized by the system when the file is uploaded into it and the system starts extracting the flagged data

### Scenario Description:

1. The system iterates the file (PDF) uploaded by the user
2. The system checks for flagged data within the file
3. The system will upload the data into an excel file

### Post Conditions: The system is ready to generate an output .csv file

### Exceptions: The system will go back to iterating the PDF file if the data it is checking is not flagged until it finds flagged data, or the document ends.

### 4.1.3 Third Use Case

### Use case name: Generate Output File

### Actors: User, System

### Preconditions:

1. File is uploaded
2. File data extracted

### Triggers: The function will be fully utilized by the system when the data is extracted from the uploaded file and the user asks the system to generate the file

### Scenario Description

1. The user clicks generate on the system
2. The system creates an output .csv file
3. The output file created includes the extracted data from the upload PDF file
4. The data is formatted in an organized manner

### Post Conditions: The user can open and view/edit the .csv file

### Exceptions: If there is no .csv file ready to be created the system should tell the user to upload a PDF first in order to extract the data and create a .csv file

### 4.1.4 Fourth Use Case

### Use case name: Open File

### Actors: User

### Preconditions: Generate output file

### Triggers: The function will be fully utilized by the system if generating the output file was successful

### Scenario Description:

1. The user will find the .csv file path after the system creates it
2. The user will open the file
3. The user can now view or edit the file as they please

### Post Conditions: None

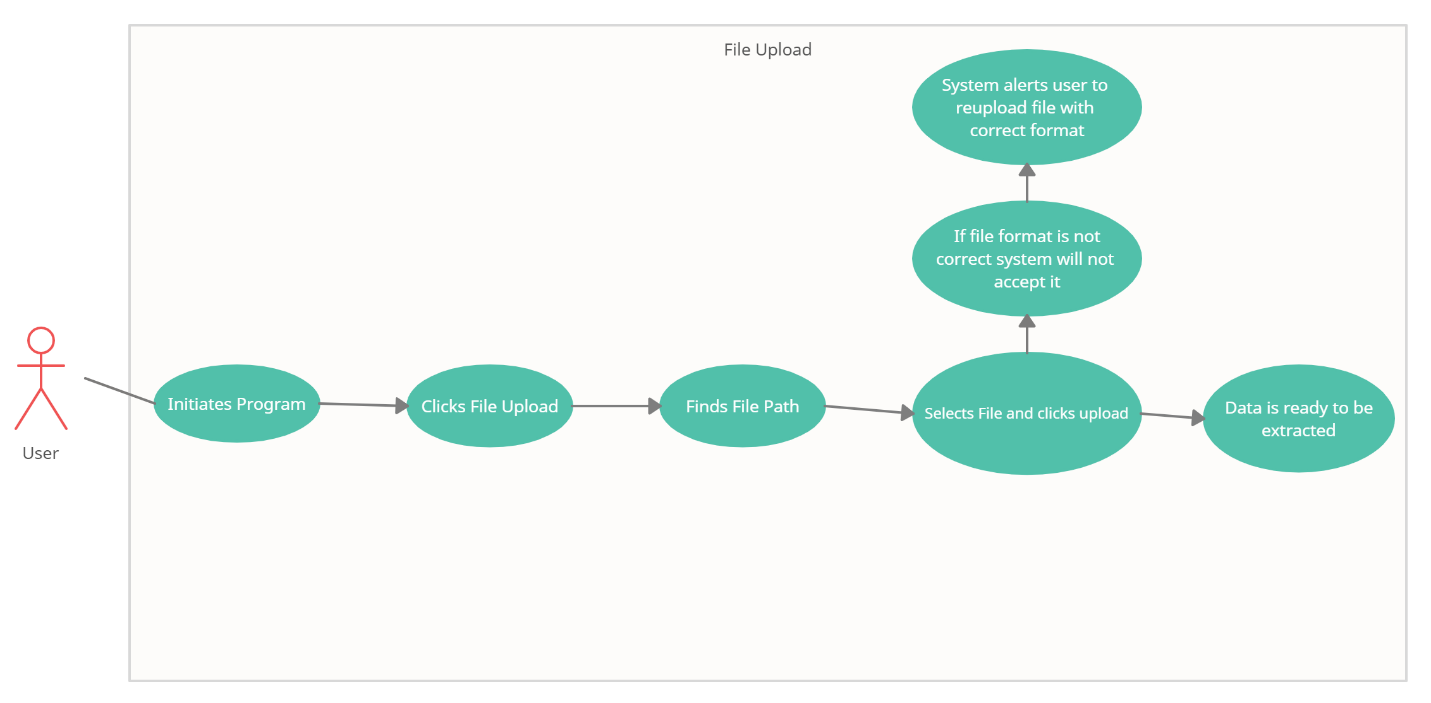
### Exceptions: None

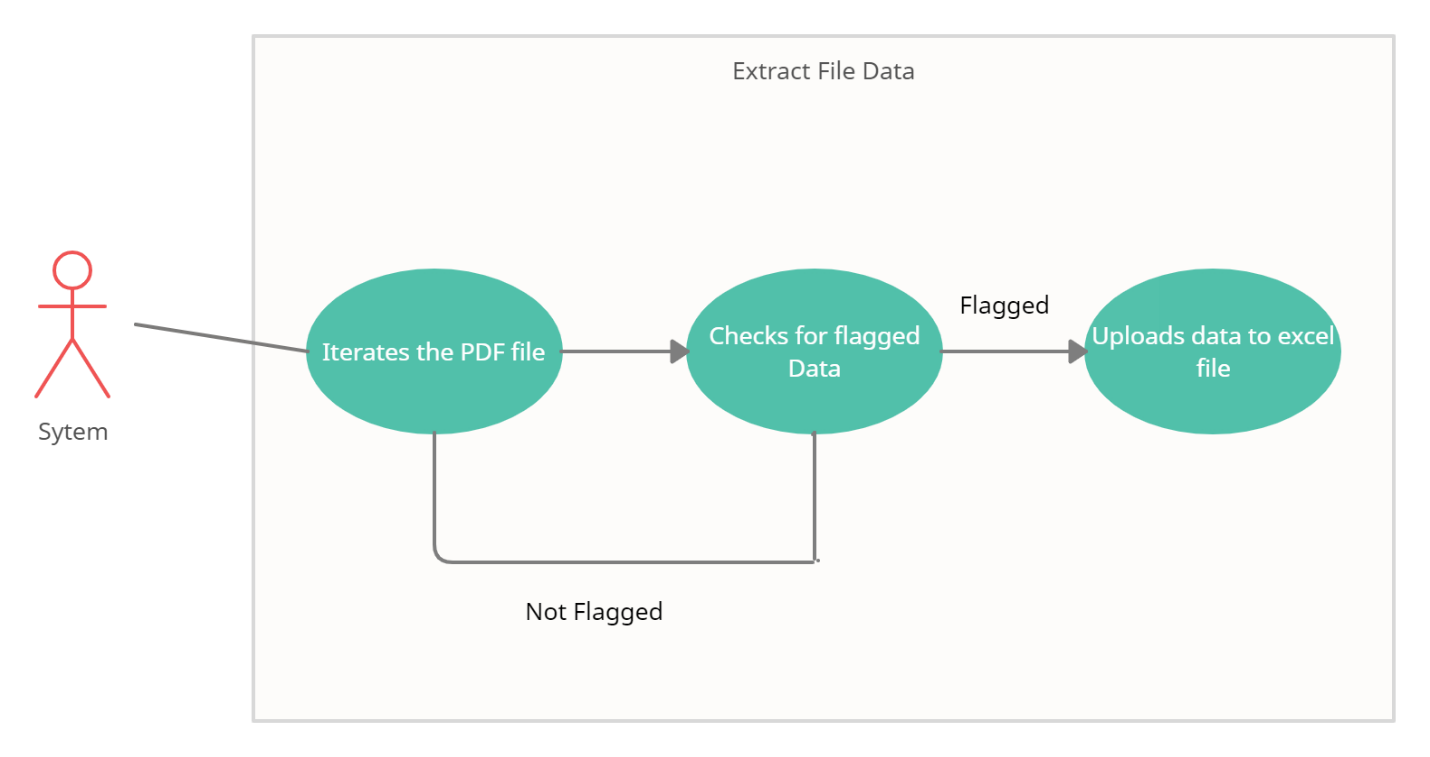
## 

## 

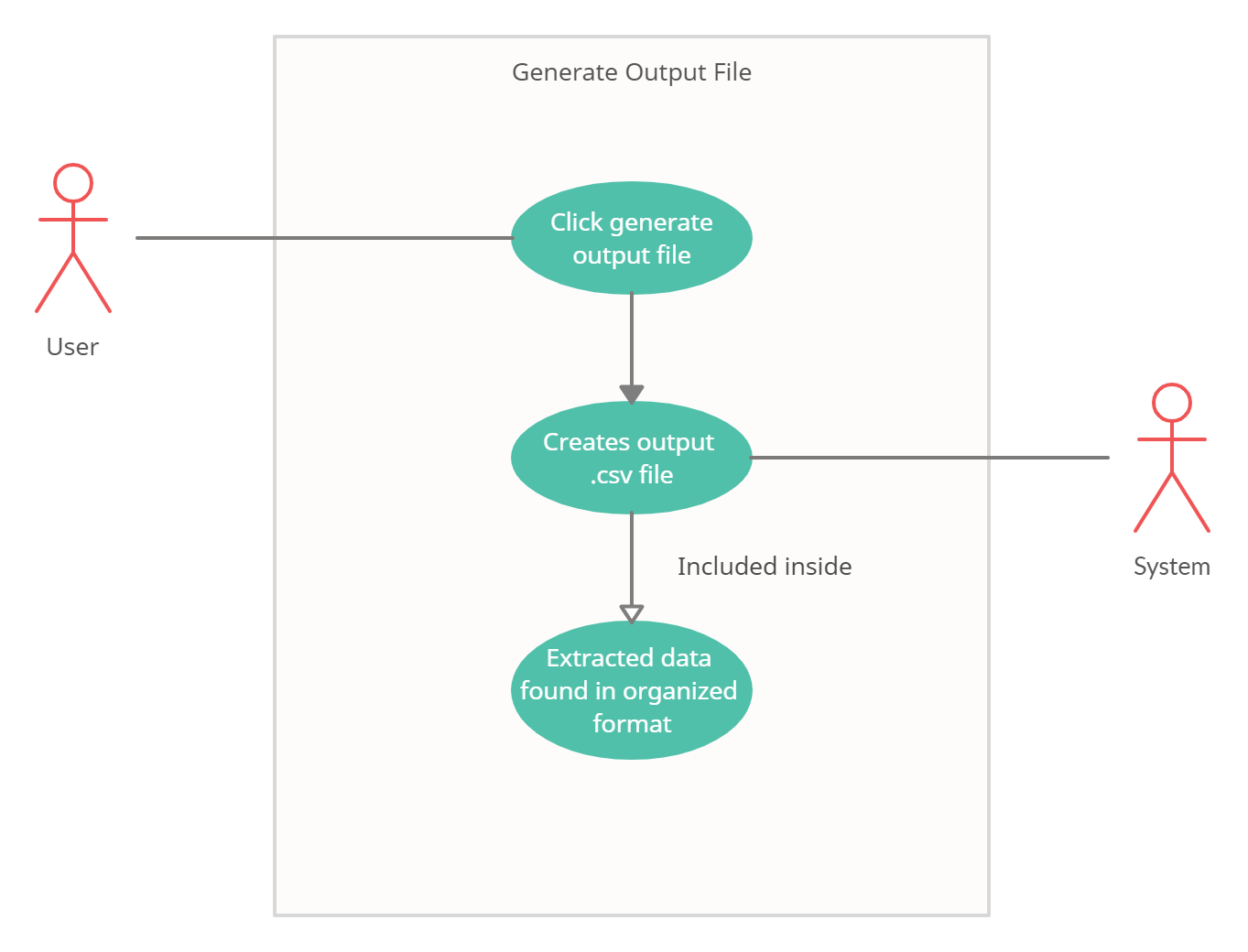
## 

## 4.3 Use Case Diagrams

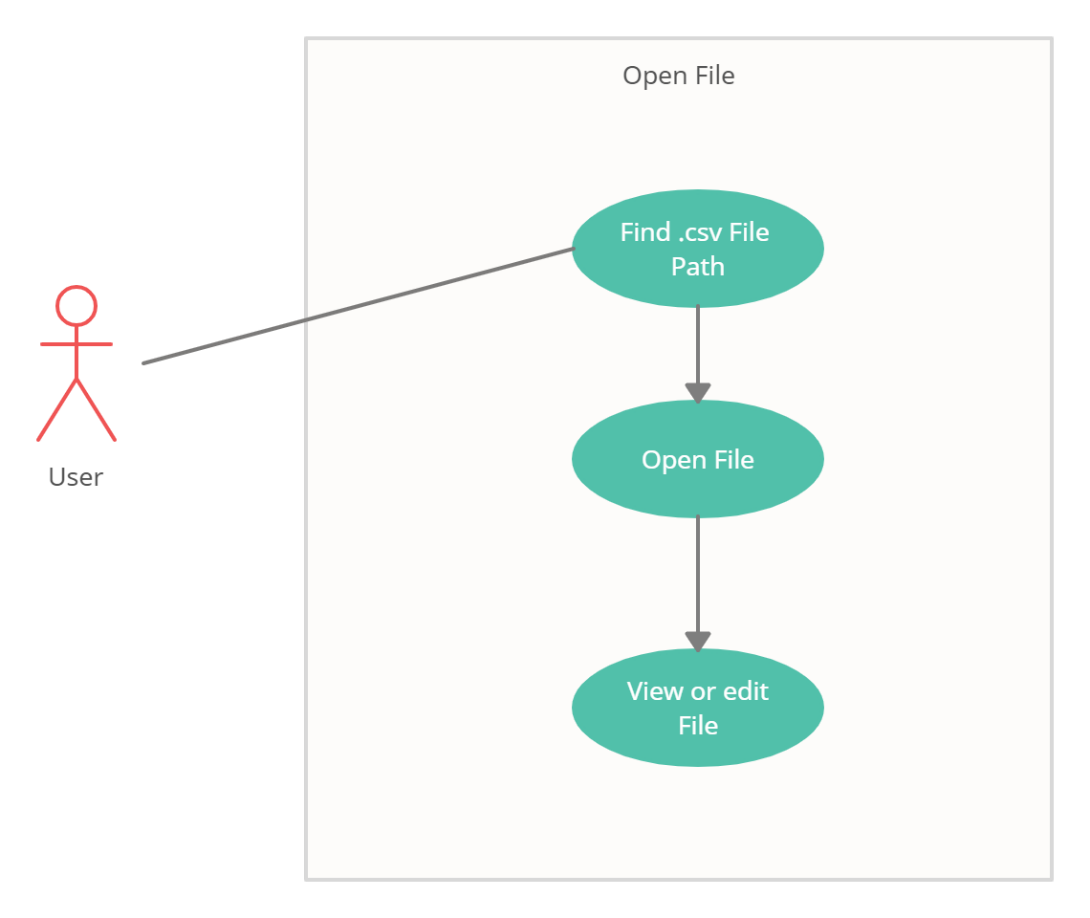
*****Figure 1 The diagram shows interaction between the user and the system during the file upload function*

****

*Figure 2 This diagram shows how the system checks for flagged data through the file*

****

*Figure 3 This figure shows the interaction between the user and the system to make an output file*

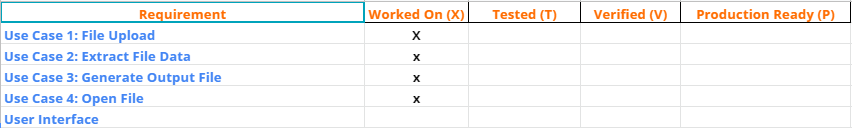
****

*Figure 4 This diagram displays how the user will open the generated file*

# 8.0 Appendices

**Presents information that supplements the Requirements Specification**

## 8.1 System traceability matrix

****

**List of References:**

-R. Klaus, J. MacFadyen, and M. Patel, “Michigan Local Government Data Parser,” 13-Aug-20AD.