Change log

- **10/24/2021**
    - Added Test plan - page 8
    - Updated Mockups - page 5
    - Added Burndown chart - page 12
    - Updated Architecture model - page 10
    - Timesheet added - page 12

Team Name: Team Runtime Terror

Team members:

Audrey Michaud - cee583@my.utsa.edu

Robert Gonzalez - yhs346@my.utsa.edu

Michael DeReus - wmi593@my.utsa.edu

Team lead: Michael DeReus

Project Name: Malicious File Analyzer

Description: Create a website that can accept and process files that need to be evaluated for maliciousness. A user should be able to query data without knowing regex, but developers may use regex or other means to process files programmatically.

Features:

a web UI (user interface)

Website will read text-based data from a user-provided URL.

Request preview of the data to be evaluated

ability to process files from the data source and extract pieces of information from the input data

Results from file processing will be viewable on the website and exportable as a CSV file

Competing Services:

Virus Total

- https://www.virustotal.com/gui/home/upload
- Analyze suspicious files and URLs to detect types of malware, automatically share them with the security community

Meta Defender

- https://metadefender.opswat.com/?lang=en
- Simply submit suspicious files to MetaDefender Cloud for analysis. A comprehensive report is created to inform you about the contents of the file.

Hybrid Analysis

- Hybrid-analysis.com
- A free malware analysis service for the community that detects and analyzes unknown threats using a unique Hybrid Analysis technology.

Target Platform:

Web Browsers

Client: NSA

Glossary

- **Subdomain**: additional part of the primary domain.
    - store.yourwebsite.com - “yourwebsite” is the domain and “store” is the subdomain. Also, “.com” is the “top-level domain”
- **Hashing**: a method of cryptography that converts any form of data into a unique string of text. Any piece of data can be hashed, no matter its size or type.
- **Bitcoin address**: Unique identifier that acts as a virtual location where bitcoin can be sent. Consists of 26-35 alphanumeric characters. This string is the public half of an asymmetric key pair. The standard format for a Bitcoin address is P2PKH (pay to public key hash)
- **Input**: a URL that points to the location of a file, local or not.
- **File** (data object): any storage object uploaded by the user for file processing.
- **File processing** (part 1): the process in which an accepted file is read by the program, and information is extracted.
- **Database**: where extracted information is stored in the form of local file references.

User Roles

Any User:

- Upload file
- View Preview
- Confirm Preview
- View analyzed file
- Export to CSV and Save to database

User Management Functional Requirements:

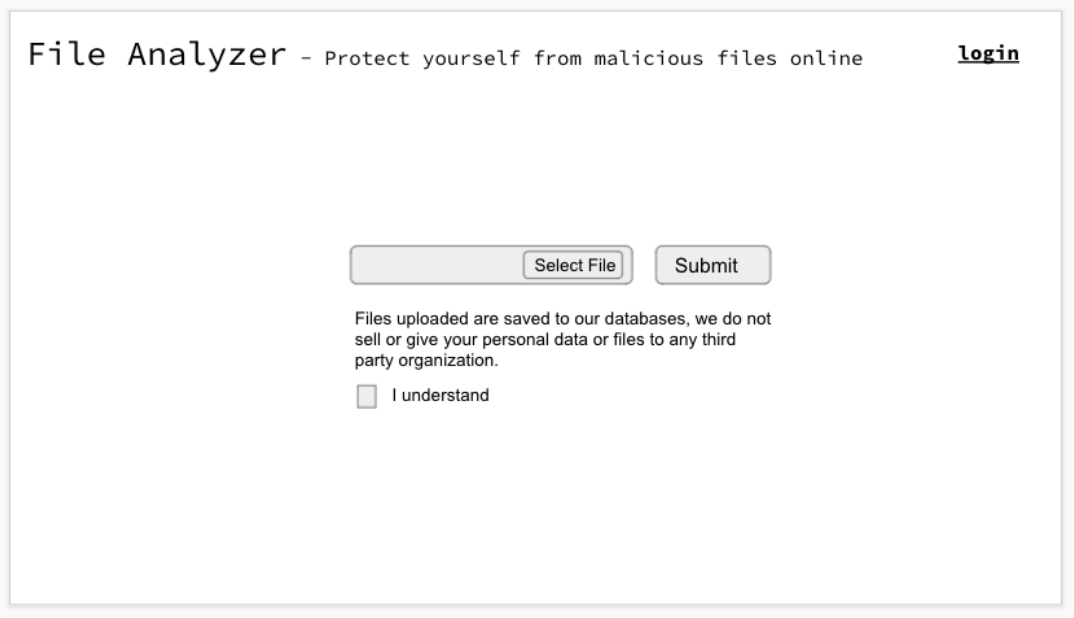
- **User uploads file**: The user is prompted to upload a file and chooses a file from a local file system.
- **User prompted to view a preview or re-enters file:** The user is asked to view a preview of the file or re-submit the file.
- **User views preview:** The user is shown a preview of the file that will be analyzed and then is asked to confirm analysis or exit and re-submit the file.
    - **User file Successfully analyzed:** The user is shown a breakdown of the file. The main information shown is the:
        - **Dates and Times**
            - Multiple formats
        - **Network information**
            - URLs/URIs
                - Hostnames
                    - IP addresses
                        - Ipv4
                        - Ipv6
                    - Domain names
                        - sub domains
                        - root domains
            - port numbers
            - protocols
            - Emails
        - **File Activity Information**
            - file names
                - Executables
                - Nonexecutables
            - file paths
            - hex addresses
            - hashes
            - cryptocurrency addresses

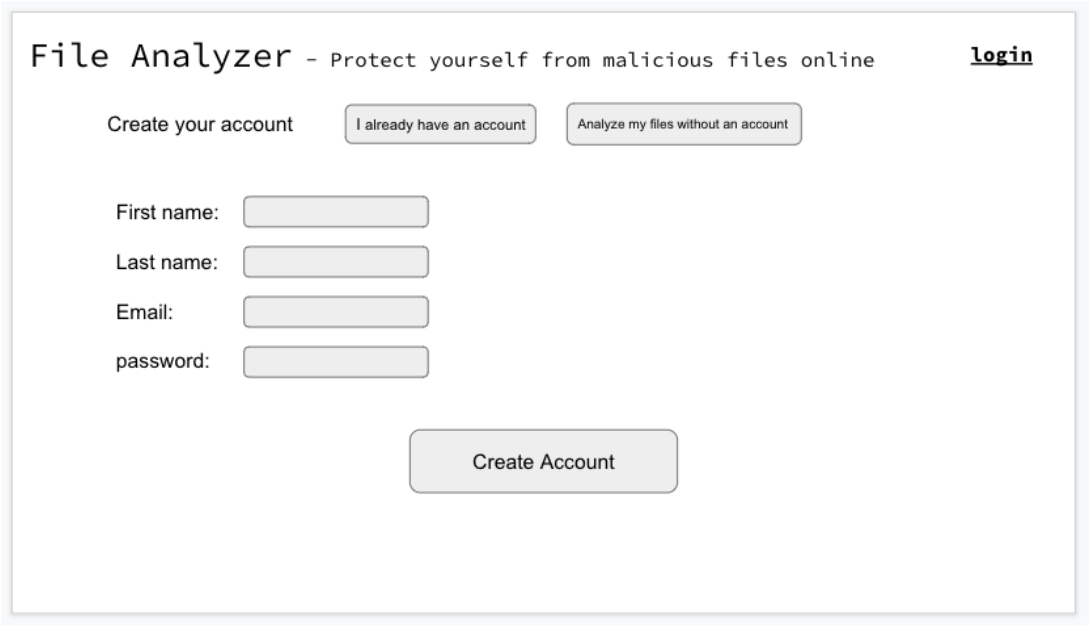
- **User export to CSV file: The user is able to export all information to a CSV file and save it on the local machine.**

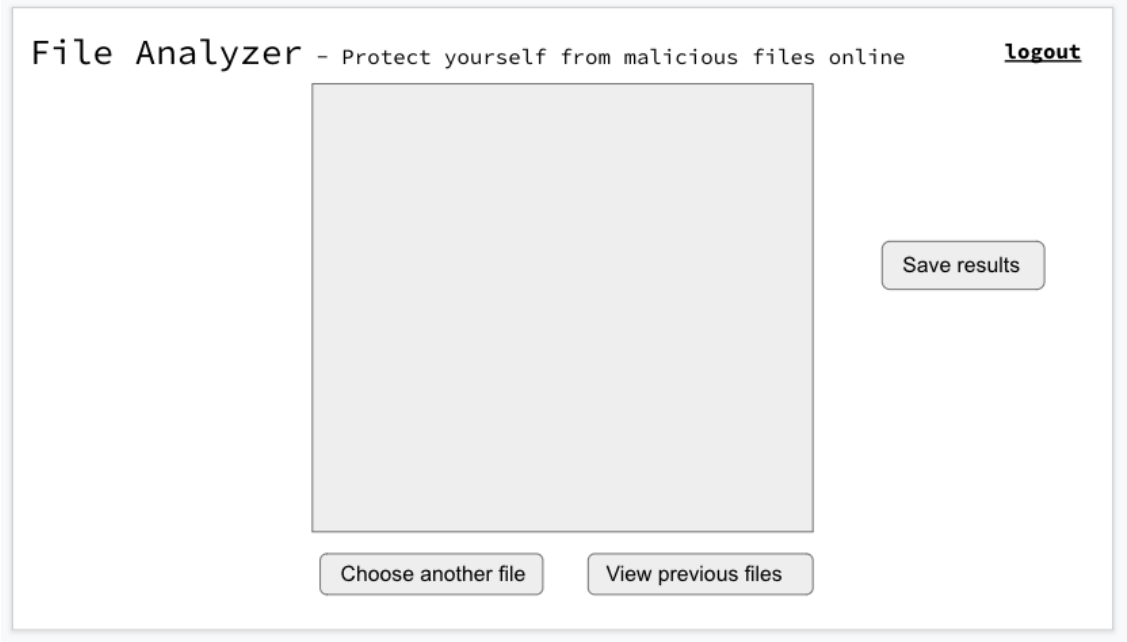
Design elements requirements

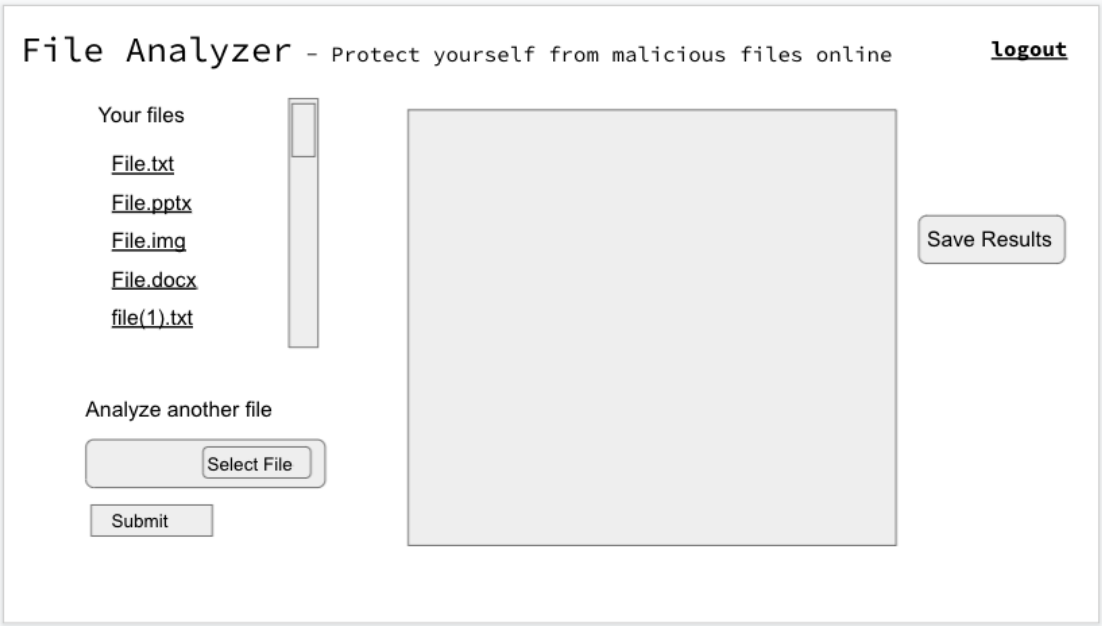
- **Database:** Mysql
- **User Interface:** Web application UI interface

UI Mockups:




File Analyzer - Protect yourself from malicious files online
login
Create your account
I already have an account
Analyze my files without an account
First name:
Last name:
Email:
password:
Create Account


File Analyzer - Protect yourself from malicious files online
logout
Save results
Choose another file
View previous files



Test Plan:

	File Analyzer								Record Test Data Values Here	EXAMPLE DATA
	File Upload									
	Name of Tester: Robert									
Test #	Test ID	Description	Expected Result	Actual Result	Test Date	Pass/Fail	Notes			
1	RU1	Upload invalid file type.	Error saying "Invalid file type please enter" (or something like that)	N/A	N/A	N/A	N/A		User File	fileEx.csv
2	RU2	Check if file was successfully uploaded	If successful continue with process. If not successful display error and try again message.	N/A	N/A	N/A			User File	
3	RU2	If logged in the file must saved and linked to the user	Data will be saved along with past data and be easily recalled	N/A	N/A	N/A	N/A			

	File Analyzer								Record Test	EXAMPLE DATA
	User Login									
	Name of Tester: Audrey									
Test #	Test ID	Description	Expected Result	Actual Resul	Test Date	Pass/Fail	Notes			
1	RU1	Login with incorrect username or password	Error saying "Invalid password or username	N/A	N/A	N/A	N/A		N/A	N/A
2	RU2	Check if username with correct password exists.	If successful continue with process and load all of the users past file uploads.	N/A	N/A	N/A			N/A	
3	RU3	User must be able to see past uploads and information about the past uploads	If logged in user must be able to see past uploads. If not logged in no past uploads are viewable	N/A	N/A	N/A				

	File Analyzer							Record Test	EXAMPLE DATA
	User Register								
	Name of Tester: Audrey								
Test #	Test ID	Description	Expected Result	Actual Result	Test Date	Pass/Fail	Notes		
1	RU1	Register without important field. Ex. Username, Password, email, ect	Error saying "Password, Name, ect is Required"	N/A	N/A	N/A	N/A	N/A	N/A
2	RU2	Make sure an identical email does not exist when creating it	If email exists don't allow user to create account and force them to use another email or login with existing email	N/A	N/A	N/A		N/A	
									M

Data Objects

INPUT (URL):

- **Data Type**: str
- **Constraints**: URL points to valid file
- **Enforced**: web UI

OUTPUT (File): Includes information about input file and other file processing results.

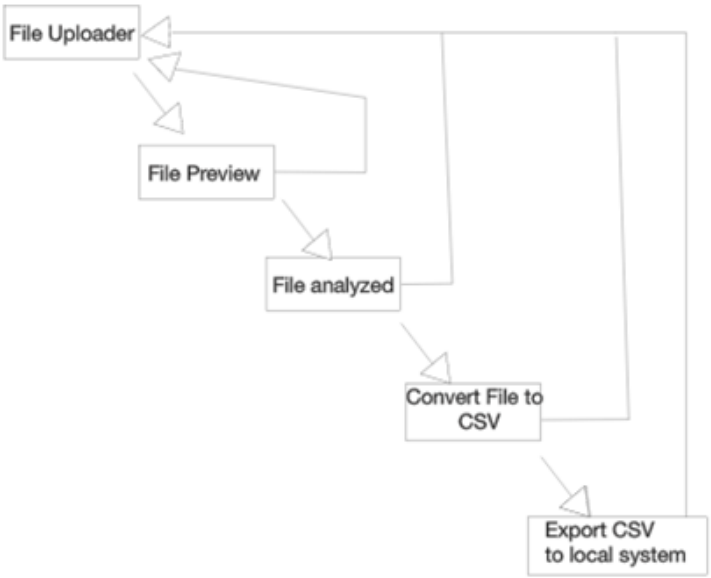
- **File type**: CSV file
- **Fields**:
    - **Dates and Times**
        - Multiple formats
    - **Network information**
        - URLs/URIs
            - Host names

        - IP addresses
            - Ipv4
            - Ipv6
        - Domain names
            - sub domains
            - root domains
    - port numbers
    - protocols
    - Emails
- **File Activity Information**
    - file names
        - Executables
        - Nonexecutables
    - file paths
    - hex addresses
    - hashes
    - cryptocurrency addresses

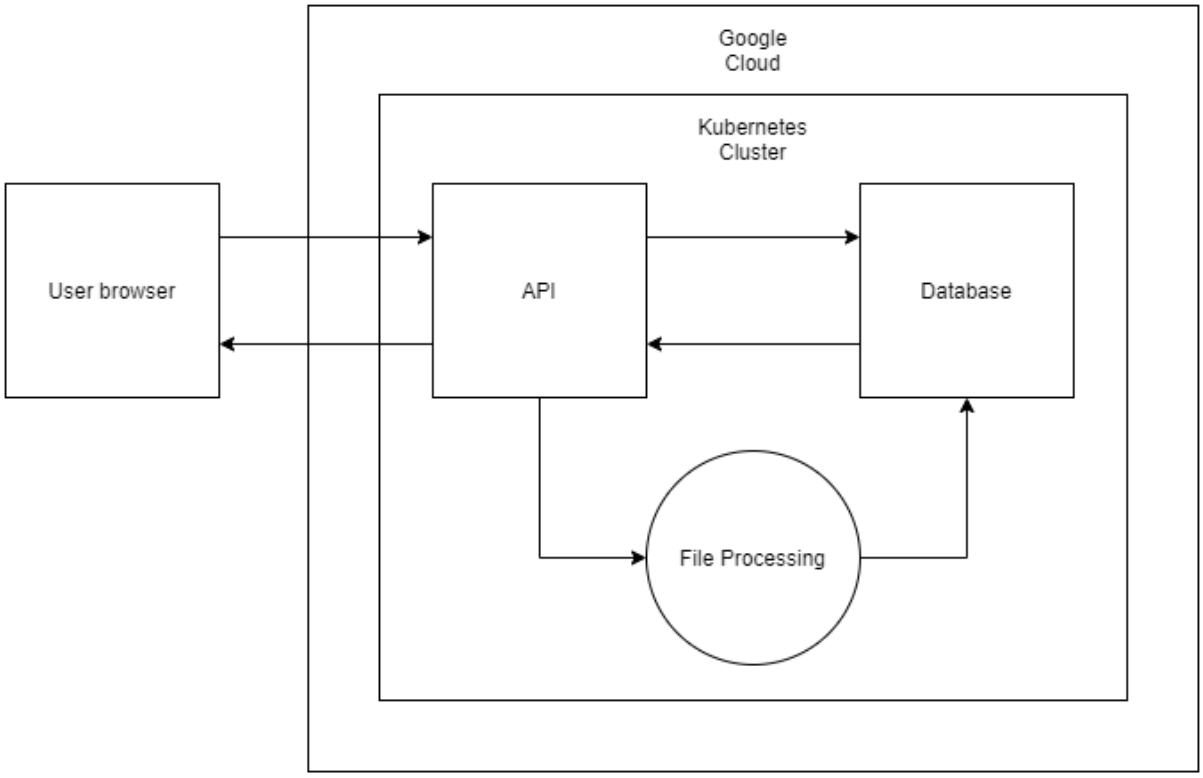
RESULT (Database Object): Location of the output file located in local storage.

- **Fields**:
    - **Input Filename**: name of original input file
        - Data Type: str
        - Constraints: not a duplicate in database
        - Primary key
    - ○ **CSV Location**: a URL path to the location of the output ".csv" file that is stored on the local file system.
        - Data Type: str

UI Navigation Flow



Architecture Model



Goals

Sprint 1

- **User Roles**
- **Define Requirements / Features**
- **Define Architecture**
- **Define Platforms**
- **Define Programming languages**
- **Milestones**
- **Wireframes/screen mockups**

Sprint 2

- **Functional database**
- **API endpoints created**

Sprint 3

- **Host Site**
- **UI mockups**

Sprint 4

- **Connect backend to database and front end**
- **File upload functionality**

Sprint 5

- **Implement heuristics to detect maliciousness**
- **Test and update**

Timesheet

Sprint 2

- Michael - 5 hours
- Audrey - 5 hours
- Rob - 5 hours

  Total: 17 hours

Sprint 3 Burndown Chart