

School of Computing and Mathematics

PRCO303  
Final Stage Computing Project

BSc (Hons) Computer Security

Shanon Fernando

File Security System  
(FileSec)

2021/2022

## Table of Contents

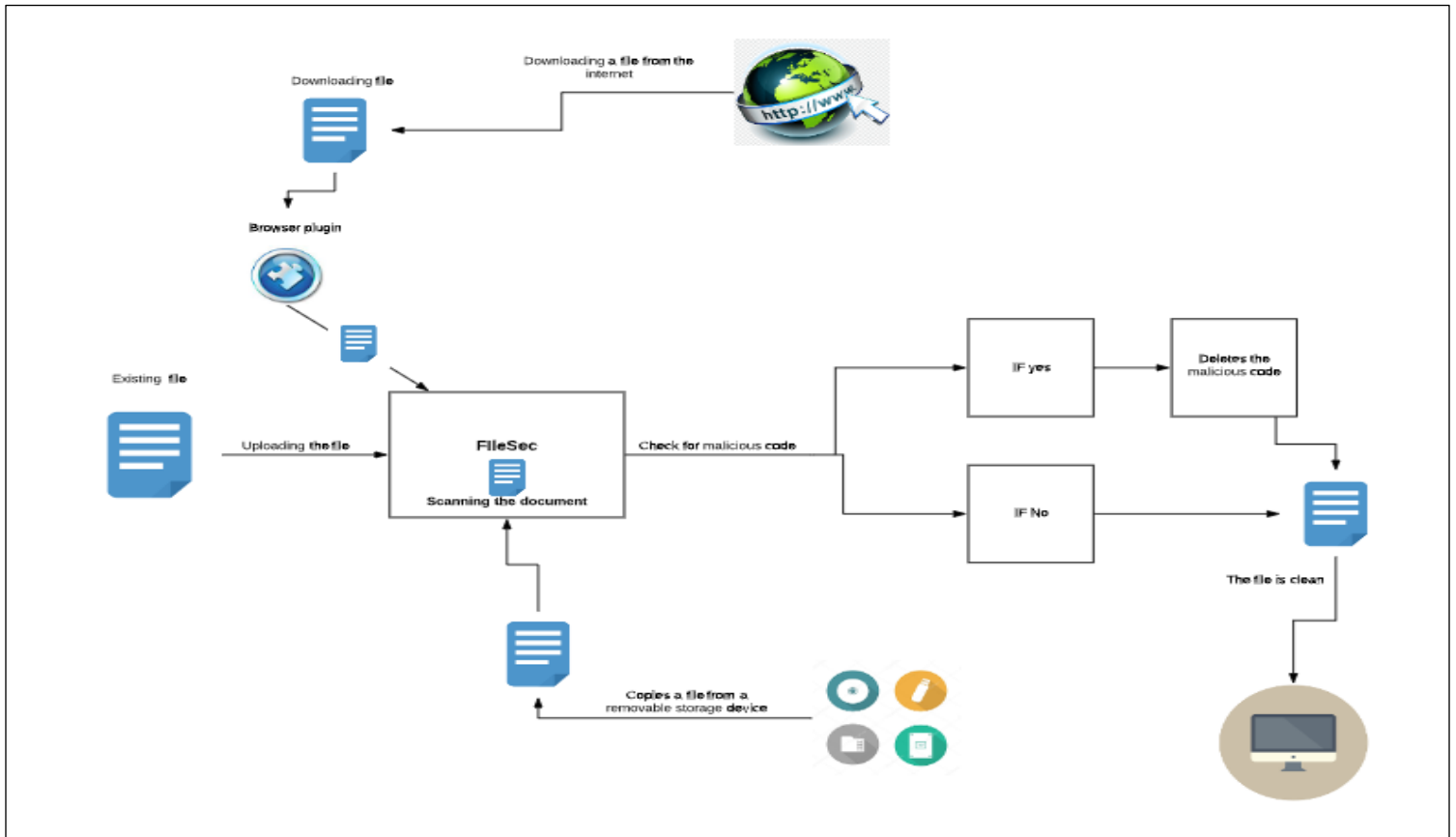
1. Tasks undertaken and outcomes .....	3
1.1 Outcome of the High-level architecture Previous .....	5
1.2 Outcome of the High-level architecture new .....	5
1.3 Outcome of the GUI development .....	10
2. Product Quality .....	12
3. Risks that have materialised and response .....	13
5. Schedule.....	14
5.1. Actual schedule (Backup schedule) .....	14
5.2. Backup schedule.....	15
6. Resources .....	16
6.1. Python Tkinter library .....	16
6.2. Virus total API .....	16
6.3. Request library .....	16
6.4. OS library .....	16
6.5. PyCharm IDE .....	16
7. Student learning undertaken and required .....	16
7.1. Python programming language .....	16
7.2. API integration .....	16
7.3. JavaScript language.....	16

## 1. Tasks undertaken and outcomes

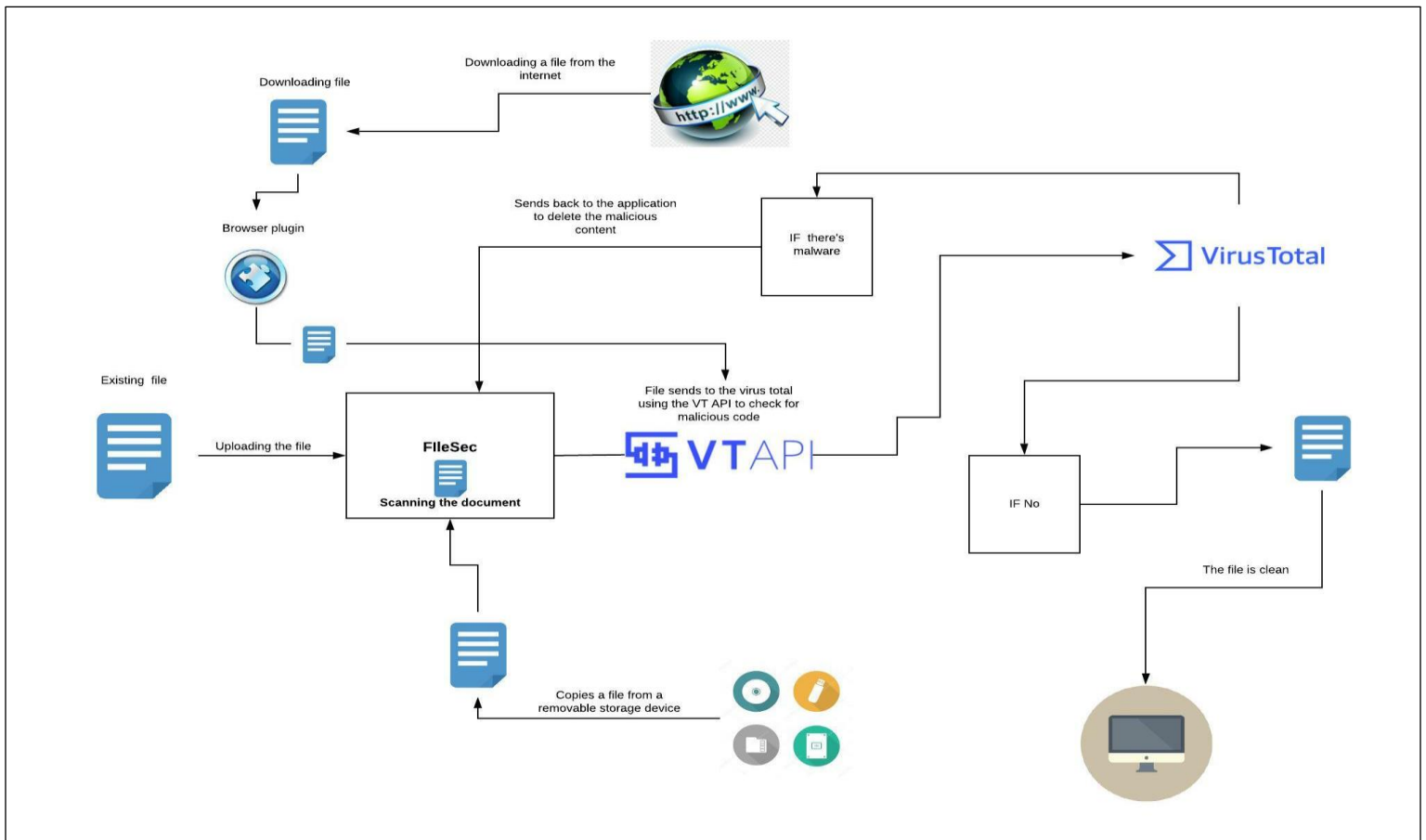
Task ID	Task	Outcome	Overall percentage
1	Requirement gathering and analysis	Gathered and analysed information and documentation related to the GUI implementation and about the Virus total API	70%
2	High level design	Changed the high-level design to a new one by integrating the virus total API (see section 1.1)	70%
3	Coding plan	No changes were done to the GUI coding plan but with the integration of the virus total API the backend coding plan was changed	70%
4	GUI implementation	GUI was implemented successfully without any designs just with a text box to display the response from the API and the upload button which takes only PDFs and Word documents to pass to the request	75%
5	Upload function and scanning	The uploading function button is working well but to scan the file and return the values the back end needs to be implemented so it's currently under development and it'll be implemented soon parallely with the browser plugin	75%
6	Implementation of the browser download manger	Implementation of the browser extension is still under development because of the unfamiliarity with the JavaScript language there's a huge learning curve when it comes to JavaScript programming, so I rate this with this 50%	50%

7	Implementation of the USB and hard drive scanner	Implementation of the USB and drive scanner is under development too for this I have created an event that will trigger when a user plugs in a USB drive or a Hard drive by using the message box widget. Since these two phases are under development I'll rate this also as 50%	50%
---	--	---	-----

## 1.1 Outcome of the High-level architecture Previous



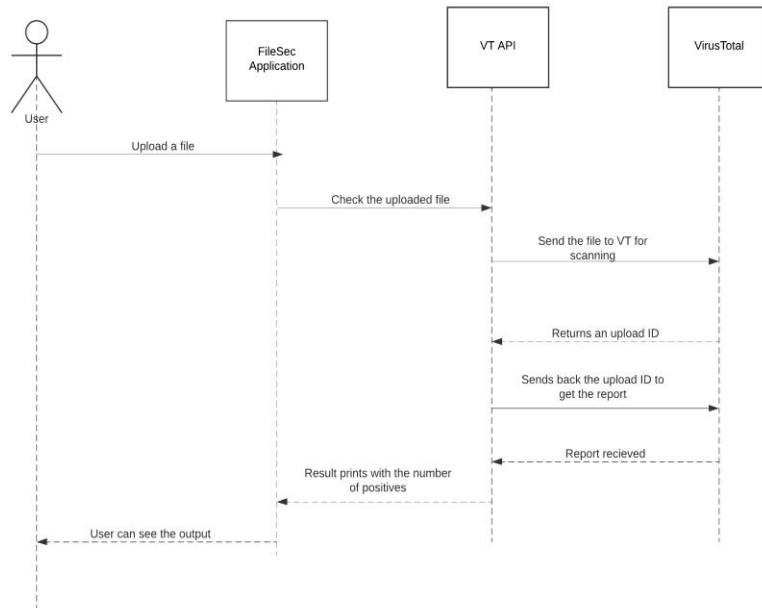
## 1.2. Outcome of the High-level architecture new



### 1.3. UML sequence diagram with each Phase individually

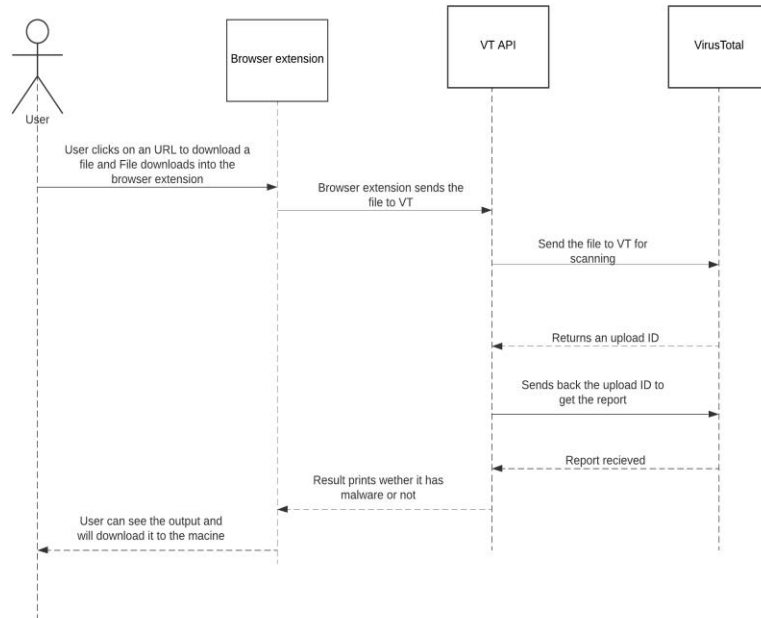
#### 1.3.1. Phase 1

##### PHASE 1



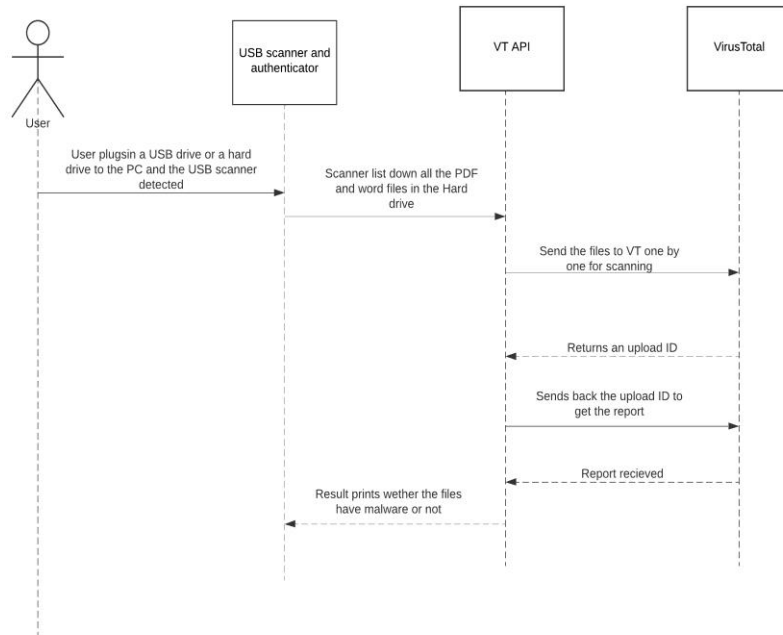
### 1.3.2. Phase 2

#### PHASE 2



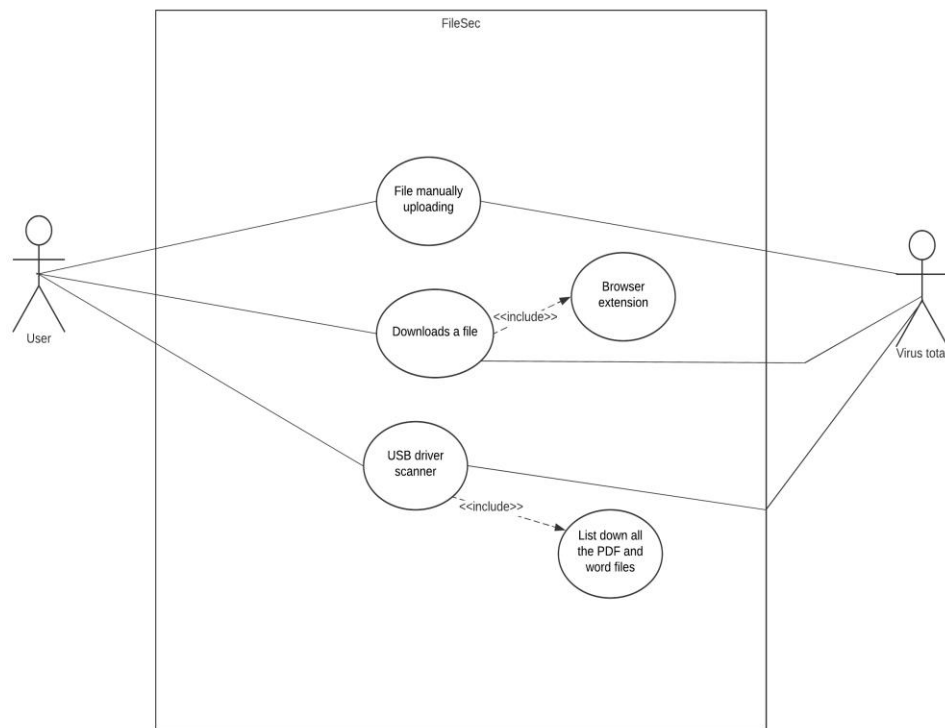
### 1.3.3. Phase 3

#### PHASE 3

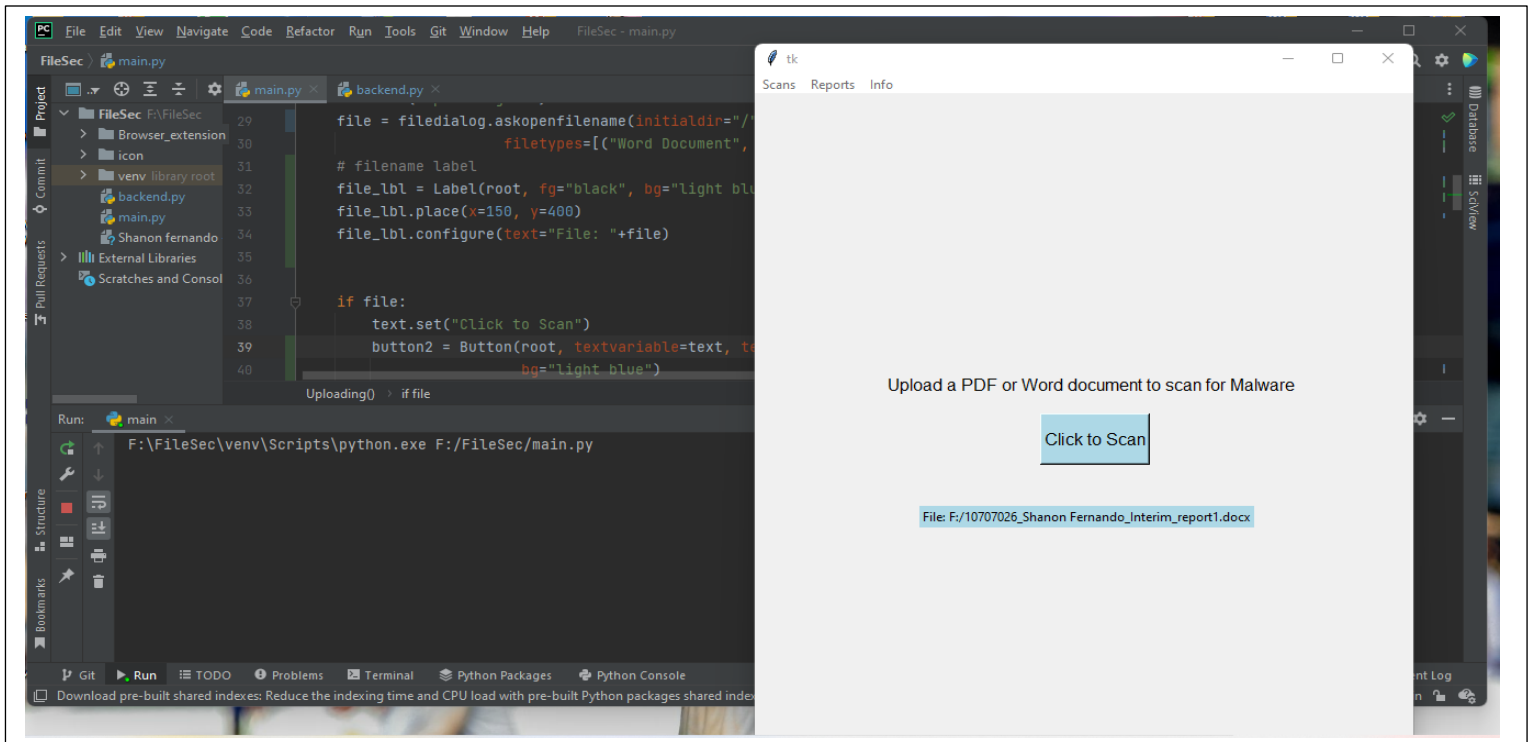




## 1.4. Use case Diagram

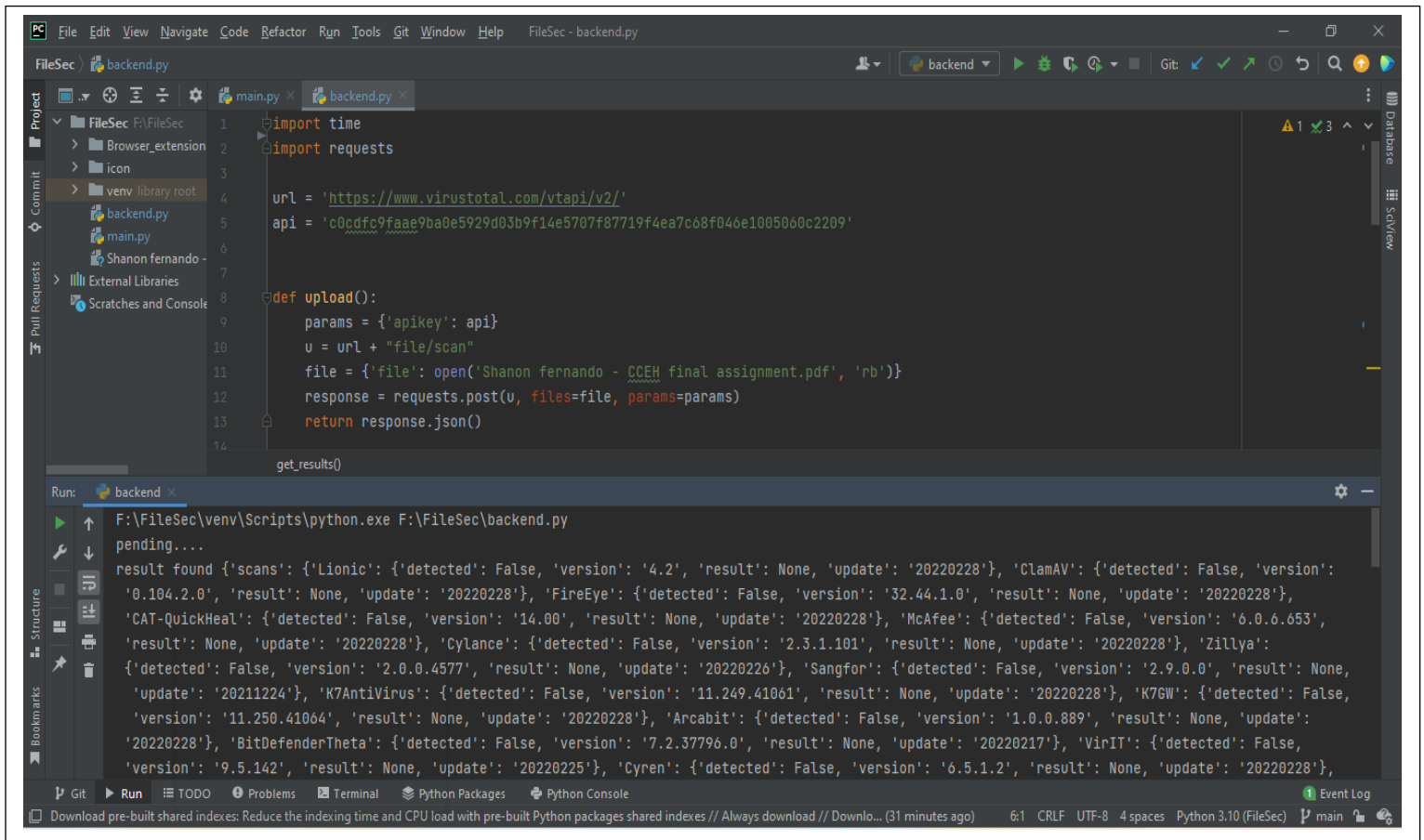


## 1.5. Outcome of the GUI development



- This is the current state of the project. The GUI of the application is completed with the basic functionalities, and it will display the selected file on the GUI.
- When you click the upload button, it asks for a PDF or a Word Document to send to VirusTotal to scan the document for any malicious content.
- The GUI and backend are developed successfully, now I have to link the backend with the GUI and print the response of the file report.

## 1.6. Outcome of the backend



The screenshot displays the PyCharm IDE interface. The main editor window shows a Python file named `backend.py` with the following code:

```
1 import time
2 import requests
3
4 url = 'https://www.virustotal.com/vtapi/v2/'
5 api = 'c0cdfc9faae9ba0e5929d03b9f14e5707f87719f4ea7c68f046e1005060c2209'
6
7
8 def upload():
9     params = {'apikey': api}
10    u = url + "file/scan"
11    file = {'file': open('Shanon fernando - CCEH final assignment.pdf', 'rb')}
12    response = requests.post(u, files=file, params=params)
13    return response.json()
14
15 get_results()
```

The Run console at the bottom shows the output of the script execution:

```
Run: backend x
F:\FileSec\venv\Scripts\python.exe F:\FileSec\backend.py
pending....
result found {'scans': {'Lionic': {'detected': False, 'version': '4.2', 'result': None, 'update': '20220228'}, 'ClamAV': {'detected': False, 'version': '0.104.2.0', 'result': None, 'update': '20220228'}, 'FireEye': {'detected': False, 'version': '32.44.1.0', 'result': None, 'update': '20220228'}, 'CAT-QuickHeal': {'detected': False, 'version': '14.00', 'result': None, 'update': '20220228'}, 'McAfee': {'detected': False, 'version': '6.0.6.653', 'result': None, 'update': '20220228'}, 'Cylance': {'detected': False, 'version': '2.3.1.101', 'result': None, 'update': '20220228'}, 'Zillya': {'detected': False, 'version': '2.0.0.4577', 'result': None, 'update': '20220226'}, 'Sangfor': {'detected': False, 'version': '2.9.0.0', 'result': None, 'update': '20211224'}, 'K7AntiVirus': {'detected': False, 'version': '11.249.41061', 'result': None, 'update': '20220228'}, 'K7GW': {'detected': False, 'version': '11.250.41064', 'result': None, 'update': '20220228'}, 'Arcabit': {'detected': False, 'version': '1.0.0.889', 'result': None, 'update': '20220228'}, 'BitDefenderTheta': {'detected': False, 'version': '7.2.37796.0', 'result': None, 'update': '20220217'}, 'VirIT': {'detected': False, 'version': '9.5.142', 'result': None, 'update': '20220225'}, 'Cyren': {'detected': False, 'version': '6.5.1.2', 'result': None, 'update': '20220228'},
```

- The backend is developed successfully but it prints the response only in the CLI, so it'll be implemented to print the response on the GUI within few days. For now, I have used a dummy file to check the functionality of the code.

## 2. Product Quality

Initial quality plan	
Quality check	Strategy
1. Requirements	Requirements will be checked accordingly in stage 2 that the gathered requirements through series of analysis process are correct.
2. Initial quality check	The initial quality check will be checked in stage 4 to ensure that the system is developed accordingly inside the initial scope.
3. Quality of the GUI	The quality of the GUI will be tested in each increment and iteration level to ensure that they full fill the requirements.
4. Quality of the functionality	The quality of the functionality will be checked in the increment level 5 and as well as in the final project stage to ensure that system functionality is as expected.
5. Quality check of the code	The quality of the code will be checked in the increment level 4 for any known bugs in the system and to comment each functionality for future needs.
6. Quality of the results	The output results from the system will be properly monitored for any false positives.

### 3. Risks that have materialised and response

Initial Risk list		
Risk	Management strategies	Responses
1. Schedule overrun	The project plan was created with some extra days to overcome this kind of risks but in some case if I miss two or three days to do conduct the processes for the allocated a backup plan to overcome this risk.	Due to the semester exams and other course work submissions, I have created backup schedule, so currently the development is going according to the backup schedule
2. Technology failures	In every development phase the system will be backed up to prevent such risks.	In every bug fixing process or when implementing new features, the code is committed to the GitHub so whenever a technology failure occurs the code can be accessible through GitHub and download it for further development.
3. Difficulty of intercepting the copying process	Extra time was allocated to learn new technologies and technique to overcome this issue.	The time for this phase has been extended to learn more techniques and technologies.
4. Huge learning curve when learning JavaScript language due to unfamiliarity.	Due to the unfamiliarity of the JavaScript language had to go through lots of documentations and videos.	Had to do some changes in the plan accordingly and developing the 2 <sup>nd</sup> and the 3 <sup>rd</sup> phases parallelly to avoid the schedule overrun.

## 5. Schedule

### 5.1. Actual schedule (Backup schedule)

6. Project plan		
Stage	Deadline	Products / Deliverables /Outcomes
Initiation	17/11	PID
Requirement gathering and analysis	21/11	Check and gather the necessary requirements to analyse how to implement the functionality of the project and to create a developer friendly working environment using GitHub.
High level design	26/11	High-level design plan is already implemented in section 5 and a high-level design of the backend is planned.
Coding plan	28/11	The coding techniques, methods, approaches, and functionalities are planned
Iteration 1	10/12	Checking the outcome of the designed graphical user interface and its status.
Increment 1	23/1	The upload function is implemented to the system.
Interim report	24/1	Submission is completed of the interim report 1.
Iteration 2	4/2	The implementation of the download manger and the browser plugin is planned.
Increment 2	1/2	The system can get any downloading file by intercepting the downloading process.
Interim 2	17/2	Submission is completed of the interim report 2.
Iteration 3	25/2	The implementation of the Driver authenticator and the interception of the file copying process is planned.
Increment 3	28/2	The system now can intercept the copying process from a removable device.
Increment 4	8/3	Check the whole system for any bugs and take note in order to fix them.
Increment 5	15/3	Round 2 testing for the bugs and verify whether the found bugs were properly fixed and the system is properly functioning.
Final Product	31/3	Round 3 testing for any bugs and a full testing of the functionality of the system.
Final Report	7/4	Submission of the final Report along with the fully functional system (PRCO303 report)

## 5.2.Backup schedule

6. Project plan		
Stage	Deadline	Products / Deliverables /Outcomes
Initiation	17/11	PID
Requirement gathering and analysis	21/11	Check and gather the necessary requirements to analyse how to implement the functionality of the project and to create a developer friendly working environment using GitHub.
High level design	26/11	High-level design plan is already implemented in section 5 and a high-level design of the backend is planned.
Coding plan	28/11	The coding techniques, methods, approaches, and functionalities are planned
Iteration 1	10/12	Checking the outcome of the designed graphical user interface and its status.
Increment 1	23/1	The upload function is implemented to the system.
Interim report	24/1	Submission is completed of the interim report 1.
Iteration 2	4/2	The implementation of the download manger and the browser plugin is planned.
Increment 2	26/2	The system can get any downloading file by intercepting the downloading process.
Interim 2	1/3	Submission is completed of the interim report 2.
Iteration 3	15/3	The implementation of the Driver authenticator and the interception of the file copying process is planned.
Increment 3	16/3	The system now can intercept the copying process from a removable device.
Increment 4	18/3	Check the whole system for any bugs and take note in order to fix them.
Increment 5	20/3	Round 2 testing for the bugs and verify whether the found bugs were properly fixed and the system is properly functioning.
Final Product	31/3	Round 3 testing for any bugs and a full testing of the functionality of the system.
Final Report	7/4	Submission of the final Report along with the fully functional system (PRCO303 report)

- I had to do some changes in the plan to cope up with the huge learning curve of the JavaScript language and developing two phases parallelly.

## 6. Resources

### 6.1. Python Tkinter library

- Python tkinter library was used to develop the GUI of the application.

### 6.2. Virus total API

- Virus total is one of the best malware scanners, so by integrating the API to my application will be able to get accurate results.

### 6.3. Request library

- The Requests library is used to make HTTP requests to the Virus total URL.

### 6.4. OS library

- The OS library used to provide functions for interacting with the operating system. Such as to go through the files in different directories.

### 6.5. PyCharm IDE

- Since the development is done using the python programming language PyCharm IDE is the most suitable and the easiest IDE to do the python development because it's developed to work with the python programming language.

## 7. Student learning undertaken and required

### 7.1. Python programming language

- Learned to program using python and learned about different python libraries and the uses of each library.

### 7.2. API integration

- Learned how to integrate, work with an API and the functionality of the API and how the requests and the responses carries up and down.

### 7.3. JavaScript language

- Learned how to program using the JavaScript language and develop browser extensions by going through lots of documentations and videos



