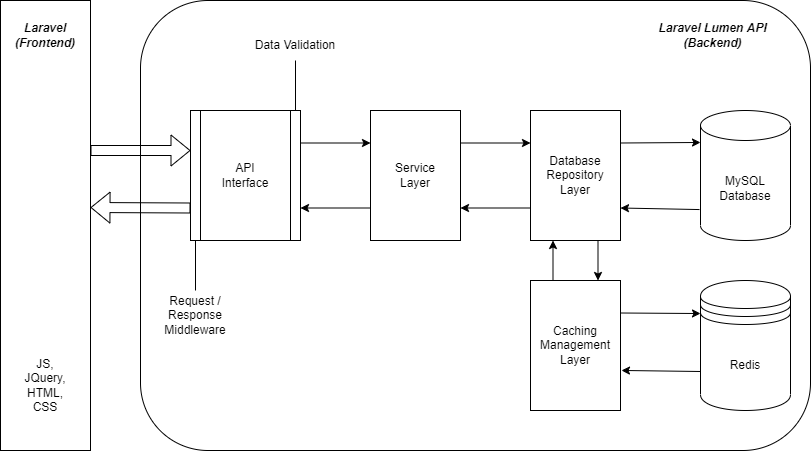
# MCCAA Tender | Digital Investigations Unit | Tech Spec

**Summary:**

The system will be developed in **Laravel** primarily. The technical architecture of the base system is the following. All users would be navigating and using the Laravel frontend interface.

**

**URL For Application: diu.mccaa.org.mt**

**MITA Appliances:**

* Azure Keyvault
* Azure Storage Container
* B2C Gov Login
* MITA Kemp
* Virtual Machine

## 1) User Roles

* Global Platform Users
  + Super Administrator
    - Read / Write to all parts of the systems
    - System Settings:
      * Adding and Revoking User Rights
      * Audit Trails
  + DIU User
    - Read/Write Access to all parts of the system **EXCEPT system settings, user rights and audit trails**
  + Chairperson
    - Read access to all functionalities
* Entity Specific Users
  + MCCAA is split up into a number of entities. Each entity has a number of directorates.
    - Each entity has a Director General with Read access to all and Write access to only that Entity
    - Each directorate has a Director / Senior Manager with Read access to all and Write access to only that directorate
    - Each directorate has a number of General Users
  + Creating, Deleting and Editing Entities and Directorates, and assigning users roles can be done only by the super administrator.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Position | System Administration Functions | Complaint Handling Function | Web Scraping / Trader Database | Safety Gate Function | Reports |
| Super Administrator | Read / Write | Read / Write | Read / Write | Read / Write | Read / Write |
| DIU User | N/A | Read / Write | Read / Write | Read / Write | Read / Write |
| Managerial User: Chairperson | N/A | Read | Read | Read | Read |
| Managerial User: Director General | N/A | Read / Write (to their respective Entity) | Read | Read | Read |
| Managerial User: Director / Senior Manager / Manager / Head | N/A | Read / Write (to their respective Entity & Directorate) | Read | Read | Read |
| General User | N/A | Read  (Individuals can be assigned Write rights if instructed by Managerial User to support investigation, on an individual case basis) | Read | Read | Read |

MCCAA’s current list of Entities, Directorates:

|  |  |
| --- | --- |
| Entity | Directorate within Entity |
| Office of the Chairperson | Quality & Improvement  Public Relations  Strategy & Implementation |
| Administration | Human Resources  Finance  IT & Digital Investigations |
| Office for Competition | Communications, Energy, Transport and Financial Services Markets Directorate  Inspectorate and Cartel Investigations Directorate  Primary, Manufacturing and Retail Markets Directorate |
| Office for Consumer Affairs | Enforcement Directorate  Complaints and Conciliation Directorate  Information, Education and Research Directorate |
| Technical Regulations Division | Market Surveillance Directorate  Regulatory Affairs Directorate |
| Standards and Metrology Institute | Standardisation Directorate  Metrology Directorate  Laboratory Services Directorate |

Example of rights with regards to Complaint Handling Function:

Complaint raised by Office for Consumer Affairs (OCA) – Enforcement Directorate: Director General (OCA) will have read / write access, while only the Director Enforcement will have Read / Write access to provide remarks and comments. The Directors of both Complaints and Conciliation Directorate and Information, Education and Research Directorate will only have read access.

Director Enforcement can then instruct the Super Admin to assign Managerial rights to a General User, within the same Directorate (not different Entity or Directorate), to the case for follow up and leave comments / remarks or provide further feedback.

## 2) Generic Functionalities

**Login Page:**

* Username & Password login page using Government CORP Account.

**Main Dashboard:**

* Front-end interface to navigate to different parts of the system according to their user role
* Menu to include the following
  + System Administration & account settings
  + Web Scraping & Data Harvesting Tool
  + Complaint Handling System
  + Safety Gate Database
  + Trader Database
* Reporting itself would not be part of the system but accessed by loading Power BI
* Complaint Handling Section
  + On main system page of the Super Admin, DIU User the cases which are pending are to be displayed, while on the main page of the Managerial User and General User, only the pending cases appertaining to the relevant Entity / Directorate and having been granted the necessary rights (in the case of the General User). These notifications will also be hyperlinks to the actual cases.

**Notification Component:**

The notification component will be receiving lists of notifications from the various components, storing them, and notifying the user of these.

Notifications to be sent by email and also on the platform.

Notifications for each component will be discussed in the component itself.

Each user will be able to turn off/on their notifications.

## 3) System Administrative Functionalities

**Audit Trail Component:**

* All reads and writes by users to the Lumen API will be recorded in the audit trail component, along with
  + Timestamp
  + User
  + Request Made

The super admin can navigate through the audit trails in the System Administrator component.

**User Management**

* Create, Promote and Delete Users with accesses
* Assign user roles to individual users
* Creation of Entities & Directorates

## 4) Web Scraping / Trader Database

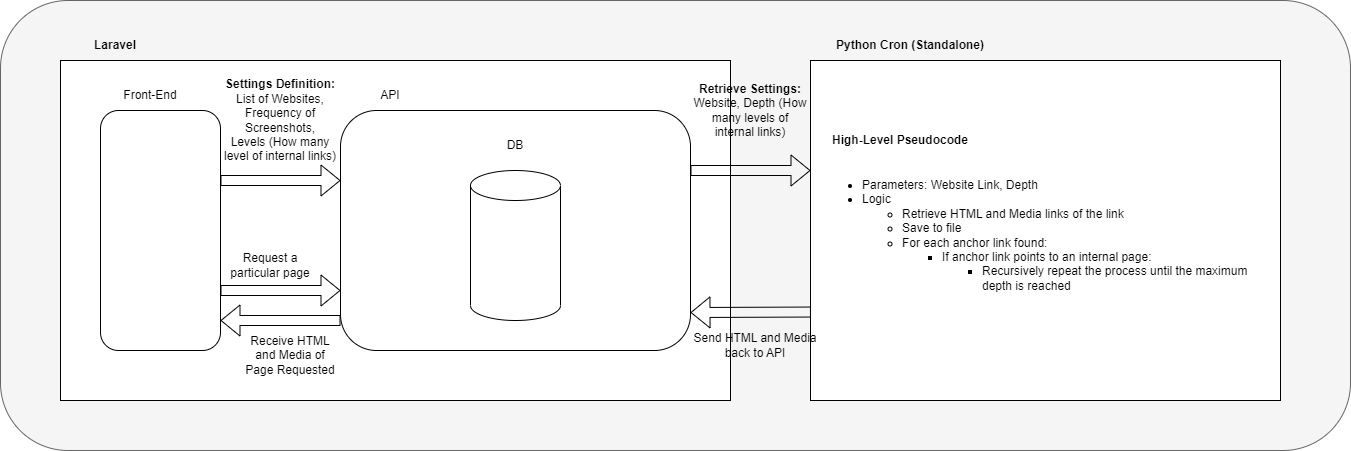
[Matthew Galea](mailto:matthew@born.mt)convert this to the new approach. Python REST API - response always OK. laravel scheduler for the next day to check success.

**Web Scraping (Crawling) Component:**

This is the most complex part of the project. From a development point of view, the script that is actually crawling the websites will be built with Python and is separate to the Laravel system.

The Python script itself is simplistic - it only generates and saves the crawled website files, but is not responsible for scheduling of crawls. That is all done by Laravel.

The diagram below also includes a high level pseudocode of the Python script:



In essence, from Laravel Frontend the “settings” of the crawler are set. The API has the necessary scheduling logic to trigger the Python script based on the settings set by the user.

The settings are the following:

* List of websites to be scraped. This can be an inner page if so wished.
* Per website, the following:
  + Frequency of crawling (how many times per week / month) should the website be scraped
  + Child links / Depth. For how many levels should the script follow the links? I.e. how many levels of subpages will it traverse.
  + Data Retention Period.

Going into more detail with regards to the crawling, the Python code has a 500MB limit. Any media (or other) files above this, are not saved to the API.

In terms of admin/user functionality, the following can be done:

* Assigning the settings including adding more websites
* Search for snapshots according to website and by date
* Link a particular snapshot to a case in the Complaint Handling System
* Viewing the scraped websites
* Change the schedule / frequency of current crawls. Either make it more frequent, less frequent, or stop it completely.
* Change the retention period of the crawled websites.

Notification when a website is no longer available or cannot be scraped. Notification only sent to DIU User & Super Admin.

**Data Harvesting Tool**

The data harvesting tool will use the HTML obtained by the Web Crawler to scrape particular information from the HTML code. The scraping itself would be done with two tools: 1) regular expressions (RegEx) and 2) ChatGPT.

In order to start the scrape, the user must select:

* A snapshot from the Web Crawler.
  + If no such snapshot exists already, it can be requested by the Data Harvesting Tool prior to starting the scraping process.
* A specific, predefined data to look for, example VAT Number
  + Or else input a RegEx or a set of RegEx which to look for

We need a list of predefined selections which MCCAA will usually look for to pre-populate the selection. Essentially this would be a list of RegEx for each data type. For example, for a VAT Number you might want to match:

* MTXXXXXXXX
* MT XXXX-XXXX
* mt XXXX XXXX

For more complex things to find, for example, Geographical location of the shop, ChatGPT will be used instead of RegEx. Each predefined data to collect will either use RegEx or ChatGPT depending on the complexity to obtain it. Another example that would use ChatGPT is checking whether a Terms and Conditions page has a “Right of Withdrawal” section.

For ChatGPT, we will feed it the HTML of the page, and ask it to identify the Geographical location. We’ll do this for all pages that form part of the crawl, until either the location is found, or else until all pages have been checked. In order for ChatGPT to have predictable outputs, we can ask it to answer in pre-defined ways.

**NB: The above assumes that ChatGPT will provide reliable output. If this isn’t the case the ChatGPT part will be removed, and the RegEx approach only is used.**

**Trader Database**

* This is a database that goes hand in hand with the Data Harvesting Tool.
* This is a table that allows for filtering, searching, sorting of trader data.
* The source of information is data gathered from the Data Harvesting Tool (scraper).
* Data can also be manually inputted / modified if needed.

## 5) Safety Gate Database

* The EU commission provides scraping tools, in particular Kibana (eSurveillance) and SAFE. Kibana: text based scraping | SAFE: reverse-image. These provide output in Excel Data.
* These Excel files can be uploaded to the system, retaining only the columns required by MCCAA
  + Week number (in the format WW-YYYY; ‘W’ denoting Week number and ‘Y’ denoting the Year)
  + Safety Gate Alert Number
  + Safety Gate Alert URL
  + Tool obtained from (Currently SAFE & eSurveillance (Kibana))
  + URL of possible matched product
* Note that a user with write access can also add further individual rows themselves
* The super admin can then do the required research and fill in other columns available in the system. These include:
  + Match Found?
  + Email to Economic Operator date
  + Reply received date
  + Reply received - Information
  + Outcome of 2nd online inspection
  + Action taken by Economic Operator
  + Follow up required?
  + Follow up email sent date
  + Follow up reply received date
  + Follow Up Reply received - Information
  + Final online inspection date
  + Any Further investigation needed?
  + Remarks
* Alerts can be shared with users to have read-only access on the system.
* Notification List (to DIU User and Super Admin):
  + 7 days after uploading the CSV file if no action has been taken by user to fill in the corresponding fields of that particular row
  + 7 days after the original inspection date if a match was found and trader was notified - i.e. “Email to Economic Operator date” has been filled
  + 14 days after original inspection date if match found, trader notified and a reminder was sent - i.e. “Follow up email sent date” filled

**NB: Notifications should be grouped by type of notification so as not to send too many notifications:**

* If when uploading the CSV file there were ten alerts and over a hundred URLs, we would not want to fill the main page with over a hundred notifications. Thus, a notification stating “There are a number of alerts from Week XX that require your attention” would suffice along with a link to the mentioned Week in the Safety Gate function.
* Same as above can be said following 7 days from original inspection date. This notification will be displayed as “There are a number of alerts from Week XX which have matched and require follow-up” along with a link to the mentioned Week in the Safety Gate function.
* Moreover, the notification following the 14 days from original inspection date. This notification will be displayed as “There are a number of alerts from Week XX that require a final inspection” along with a link to the mentioned Week in the Safety Gate function.

## 6) Complaint Handling System

This is essentially a database of complaints.

* Allows creation, modification and deletion of complaints
* Allows searching, filtering and sorting of complaints
* Expected fields:
  + Ref No.
  + Request Date
  + Entity
  + Directorate
  + Description
  + Comments
  + Status
  + Date Closed
* Apart from the above, each case is assigned a number of users:
  + Users with full access to edit complaint
  + Users with access to add feedback only
  + Users with read-only access
* The assignee of users towards complaints is the super admin
* Also allows linking of scraped data in Web Crawler component
* Notifications
  + When a new case is opened and assigned to a DIU user (Notification received by the assigned DIU User and the Managerial User depending on the Entity / Directorate the case was opened on).
  + When a case has been reassigned to another user (Notification received by the newly assigned DIU User).
  + When a General User is given access to a specific case as instructed by the respective Entity / Directorate Head (Notification received by the Super Admin, The Managerial User, and the concerning General User).
  + When a case has been opened for 15 days and no change in status has occurred (Notification received by the assigned DIU User and the Super Admin).
  + When a case has been opened for 30 days and status is not set as closed (Notification received by the assigned DIU User and the Super Admin).
  + When a case has been closed (Notification received by the assigned DIU User, the Super Admin, and the Managerial User depending on the Entity / Directorate the case was opened on, and, when applicable, the General User given necessary rights to comment on the case).
  + When a case is reopened and assigned (Notification received by the assigned DIU User and the Managerial User depending on the Entity / Directorate the case was opened on).

## 7) Reporting

Reporting will be completely handled by Power BI Desktop application. We will set-up predefined dashboards according to the contractor’s needs. The easy to use drag and drop editor will allow the contractor to create further dashboards as needed.

PowerBI would retrieve the data from the SQL database. The IPs of the users that can access the SQL data with Power BI would need to be whitelisted.