Deliverable 3.3
DIAGNOSTIC IMAGE REPOSITORY

Grant Agreement:	862848
Project Title:	Linking East and West African farming systems experience into a BELT of Sustainable Intensification
Project Acronym:	EWA-BELT
Project Start Date:	1st October 2020
Related work package:	Work Package 3
Lead Beneficiary:	OCCAM
Submission date:	31st March 2022
Nature:	Website, patents filling, etc.
Dissemination Level:	Public

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 862848

DISCLAIMER

ACKNOWLEDGEMENT

This document is a deliverable of the EWA-BELT project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement Number 862848.

Authors: Simone Merafina*, Robert Barbieri*.

Author Affiliation:

*OCCAM, Osservatorio sulla Comunicazione Culturale e Audiovisiva nel Mediterraneo e nel Mondo

Correspondence:

* Via Duccio da Boninsegna, 21, 20145, Milan, Italy. occam@occam.org

TABLE OF CONTENTS

Sommario

List of Acronyms

AI: Artificial Intelligence

CRUD: Create, Read, Update, Delete

DB: Database

DBMS: Database Management System

DS: District Service

FFRUs: Farmer Field Research Units

IoT: Internet of Things

NC: National Center

ORM: Object–relational mapping

PHP: Hypertext Preprocessor

qPCR: quantitative polymerase chain reaction

UML: Unified Modeling Language

URL: Uniform Resource Locator

Executive Summary

The Task 3.1 aims at implementing a remote PLANT HEAlth Diagnostic (PLANTHEAD) network in Africa to promote real time diagnosis and environment-friendly crop protection approaches in resource-constrained environments lacking the organizational and/or the sociotechnical system resources to cope with food insecurity issues.
The proposed scheme involves an authenticated End User (grower), who inserts the request information through a mobile App, takes a georeferenced picture of the symptoms, submits the request to the first hub (e.g., the District Service, DS). The DS validates the request, inserts further notes/attachments, provides initial feedback, and submits the request to the National Centre (e.g., the main University or Research Center, NC). The National Center analyses the request, monitors the environmental situation, validates the request, and inserts further notes/attachments, providing additional feedback and, if needed, submits the request to the Service Provider (represented by a collaborative network of universities and research institutions with renowned expertise in plant health, epidemiology, plant disease and pest control). The Service Provider analyses the problem, provides a diagnosis and control feedback, sends an initial solution to the National Centres (and through them, to the District Services and back to the End User), shares the results that will update a knowledge base for intelligence services (AI-Machine Learning).
The creation of a Platform and a database for data analysis and model creation, enriched by all the research results of the various work packages, will provide useful tools available to researchers, professors, and scientists. All the elements developed during the project will be stored within our Platform, in the appropriate repository. The storage of images in the repository represents the fulcrum of the development of the diagnosis in real time, as it will create the underpins from which the trained model of Artificial Intelligence will be developed.

The Diagnostic Image Repository

1.1 The conceptualization of the Image Repository

The image repository (reachable by accessing the platform from the following link: https://app.ewabelt.eu/) represents one of the most important functions of the PLANTHEAD Platform. The repository is connected to the database, allowing on one hand to store image files (that will be used for the plant diseases recognition), and on the other hand to store of all other types of files, enhancing the creation of a mutual information exchange network, creating the knowledge-belt between East and West Africa.
Therefore, other than the collected images for the diagnosis, the knowledge emerging from all the tasks of the project will be stored in the PLANTHEAD Repository and it will be easily accessible to all the farmers involved and disseminated through different and appropriated communication channels.
From a technical point of view, during the conceptualization of the repository, it was decided that:

- the Aruba will be used as a server provider and as Operating System will be used Linux.
- The programming language will be PHP (latest version) and as framework Symfony will be used.
- Maria DB will be used as Database Management System.

OCCAM proceeded with the design of all the possible applications contained in the Platform. It has been analysed in terms of costs, time and results to be achieved.

1.2 The Development of the Image Repository

Respecting the principle of the best value for money, OCCAM selected IT professionals with an expertise in the Platform development using the Symfony framework. The programming and development work was divided into several Phases, with the respective Tasks.
The following Tasks emerged for the Phase 1:

1. Configuration of the development environment, creation of the repository and configuration of the deploy procedure for implementing safe start-up on the host server.
2. Installation and configuration of the PHP Symfony Framework and integration with the deploy procedure.
3. Configuration of the graphic template within the Symfony Framework.
4. Configuration and management of users with different roles, authentication methods and system functionalities.

5. Creation of role-based differentiated menus.

The following Tasks emerged for the Phase 2:

1. Creation of the whole of CRUD to add, edit, delete and view the system entities. The creation of the CRUD consists in the creation of all the screens and all the forms for inserting, editing and deleting all the elements of the system, therefore users, nodes, WPs and tickets. Hence, all the forms will be created with the necessary fields and the screens that will allow admin users to manage everything via the admin panel, including insertion, update and deletion.
2. Management and creation of the ticketing system, i.e. a registry created within the Platform that shall include all the requests for assistance sent by farmers (three alert modes with management of request at different nodes).
3. Possibility to manage nodes based on geolocalization with standard configuration and subsequent manual edits based on needs.
4. Creation of a panel to create association to different forms, offering a direct link to users to fill out the survey. The system will offer a mapped survey list with the direct URL to the KoboToolbox.

1.3 Technical specifications

Services	Version	Use
NGNIX	1.10	Web server
PHP	8.1	Server-side programming language
MariaDB	10.7.3	Database Server
Symfony	5.0	PHP framework used to build the application
Javascript		Client-side programming language
jQuery	3.0	Javascript library
HTML	-	-
CSS	-	-
Bootstrap	4	CSS Framework

Services	Use
Google Maps	Geocoding of the city in coordinates
IpStack	Geocoding service through IP address
Mailgun	Email provider service

Technical specification of the hardware
Shared VPS
Debian9 64bit
2vCPU
4gb RAM
10gb Hard Disk

1.4 Structure of the Image Repository

The PLANTHEAD Platform contains multiple functions. The repository represents one of the most impactful functions for all the partners of the EWA-BELT Consortium. Not only does the Repository contain all the images to develop a real-time diagnostic system (Image Repository), but it can contain all the files that may be needed during the life of the project. The repository, directly connected to the database and the cloud server of the Platform located in Italy, is sorted in folders, where each folder contains the respective elements. As previous said, the repository interacts with the database. The repository allows:

- The management of data, which then allow you to create the respective folders and to put all the results of the research (files .txt, .pdf, .jpg, .png, .xls, .xml, etc.).

- The management of alert processes, where it is possible to see the picture sent by the farmers and the comments of various nodes.

- The possibility to create folders and subfolders, in turn add files and modify their access level through the roles created in the system.

- The creation of a folder with all the pictures collected, to create a phytopathological recognition system that uses neural networks and AI, which in turn will use the photos in the folders for machine learning.

- Store different types of files and make them accessible based on different user roles, thus allowing various levels of access

Technical explanation of the ways of interaction between the repository and the Platform: It is possible to access the repository only through credentials and the interaction with the application is developed using PHP source code with ORM Doctrine of the Symfony framework used to build the Platform.

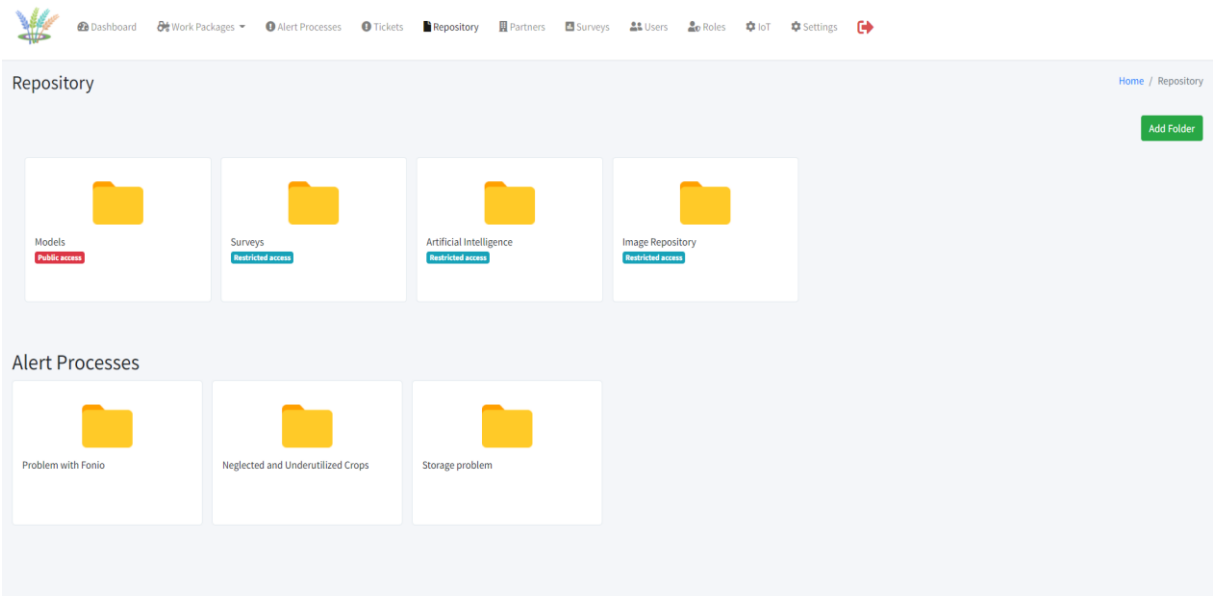

Table 1: Example of the frontend picture of the repository

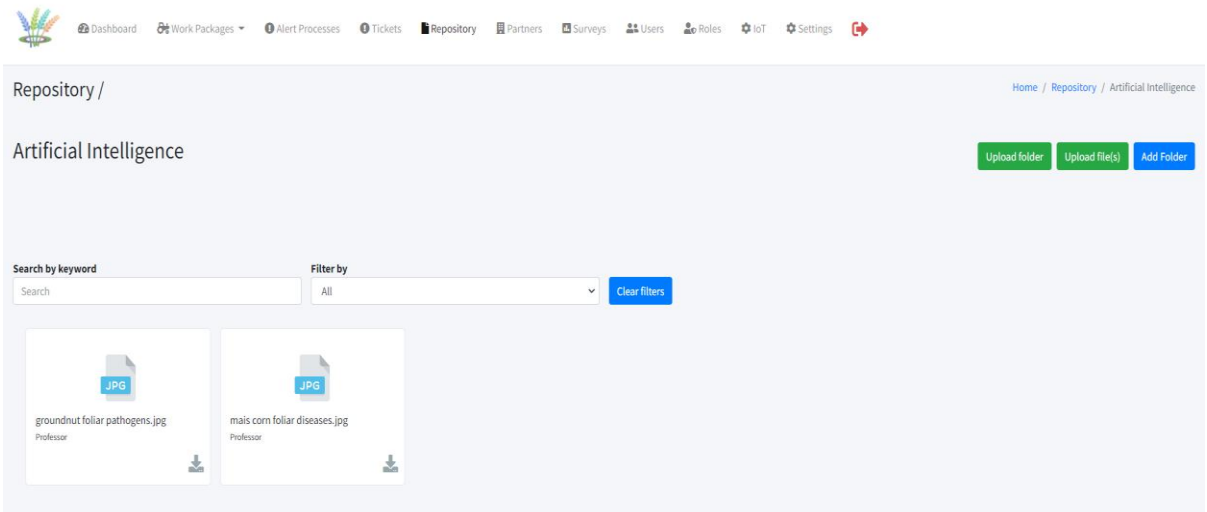

Table 2: Example of the frontend picture of the repository

```php
<?php

namespace App\Repository;

use ...

class AlertProcessRepository extends ServiceEntityRepository
{
    public function __construct(ManagerRegistry $registry)
    {
        parent::__construct($registry, entityClass: AlertProcess::class);
    }


    public function findByTickets($tickets)
    {


        return $this->createQueryBuilder( alias: 'a')
            ->where( predicates: 'a.tickets IN (:tickets)')
            ->setParameter( key: 'tickets', $tickets)
            ->getQuery()
            ->getResult();
    }


}
```

Table 3: Example of the backend pictures of the source code

```php
public function getFoldersFromUser(User $user, $parent)
{
    $workPackages = $user->getWorkPackages();
    $roles = $user->getRoles();
    $role = reset( &array: $roles);


    //Get all folder ids where user is allowed
    $query = $this->createQueryBuilder( alias: 'f');
    $query = $query->select( select: 'f.id')
        ->innerJoin( join: 'f.users', alias: "user")
        ->andWhere("user.id = :user")
        ->setParameter( key: 'user', $user->getId());

    if ($parent) {
        $query = $query->andWhere('f.parentFolder = :parent')
            ->setParameter( key: 'parent', $parent);
    } else {
        $query = $query->andWhere('f.parentFolder IS NULL');
    }

    $folderIds = $query->getQuery()
        ->getArrayResult();

    $folderIds = array_map(function ($folder) {
        return $folder['id'];
    }, $folderIds);

    $folderIds = array_values($folderIds);

    // check folders where are public or have permissions
    $query = $this->createQueryBuilder( alias: 'f');
```

Table 4: Example of the backend pictures of the source code

1.5 Repository management and further development

To facilitate the repository management, OCCAM will lead a training in order to allow the management of the repository within the platform. Updates will be added on at least a fortnightly basis or more frequently depending on activities of the Consortium.