Internship project report

Belguebli Rayane

Table of Contents

Abstract

Our project aims to create an innovative web platform that acts as a management centre for maintenance tasks. Data from a CMMS relating to tasks to be carried out, planned by managers, is retrieved and presented interactively to technicians via glasses equipped with AR technology. This information, integrated directly into their field of vision, gives technicians rapid access to instructions and task details, improving their efficiency and accuracy when working in the field. By using Django based on the MVT model, we are ensuring a robust and scalable architecture to support this complex integration between maintenance data and AR technology.

Chapter 1

Introduction

In the field of industrial maintenance, the efficient management of tasks and resources is crucial to ensuring the productivity and reliability of operations (Malta, Mendes, and Farinha 2021) [1]. Our project addresses this issue by proposing an innovative solution that exploits emerging technologies to facilitate the maintenance process. Inspired by recent advances on ICMMS (Fu et al. 2002) [2], we aim to design an integrated Django-based web platform to orchestrate coordination between managers and technicians, while integrating Augmented Reality (AR) features for an immersive experience.

This solution meets a growing need in the field of industrial maintenance, where the complexity of equipment and operations requires agile and efficient management. By combining the power of asset management with the ease of use of an intuitive web interface, our solution aims to streamline maintenance processes while providing an optimal user experience.

Our aim is to provide a versatile and adaptable platform that can be tailored to the specific needs of different industries and businesses. Through the integration of AR, technicians will be able to access contextual information in real time, improving their efficiency and accuracy when working in the field.

This project represents an exciting opportunity for an internship abroad, offering the chance to explore new technologies and contribute to innovative solutions in the field of industrial maintenance. By taking on this challenge, we aim to acquire new skills.

Chapter 2

What is CMMS ?

A Computerised Maintenance Management System (CMMS) is an essential tool for the efficient management of maintenance activities within an organisation. Whether in industry, services or public institutions, a CMMS provides a centralised platform where maintenance teams can effectively plan, execute and monitor their tasks.

One of the key benefits of a CMMS is its ability to integrate new technologies such as mobility and traceability applications. Using mobile devices such as smartphones or tablets, field technicians can access maintenance information in real time, enter data on the spot and receive instant instructions. This significantly improves the responsiveness and efficiency of the maintenance team, reducing unplanned downtime.

In addition, traceability of maintenance activities is a crucial element in ensuring regulatory compliance and optimising processes. Modern CMMSs offer advanced monitoring and reporting capabilities, enabling managers to track maintenance histories, analyse trends and make informed decisions to improve overall asset performance.

Studies and articles have been published to demonstrate the effectiveness of CMMS in different sectors. For example, research carried out by experts in maintenance management revealed that the implementation of a CMMS in an industrial company led to a significant reduction in downtime and maintenance costs, while improving equipment availability (Shankar, Singh, and Singh 2021) [3].

Similarly, another case study that highlighted the benefits of a CMMS in the utilities sector, showing how a municipality was able to optimise its public infrastructure maintenance operations through the use of an integrated CMMS solution (Kour et al. 2022) [4].

In summary, CMMS are essential tools for maintenance teams, providing efficient management of maintenance activities while taking advantage of new technologies to improve responsiveness and traceability. Studies and research articles support the positive impact of CMMS in different sectors, highlighting their value as a key element of asset management and preventive maintenance.

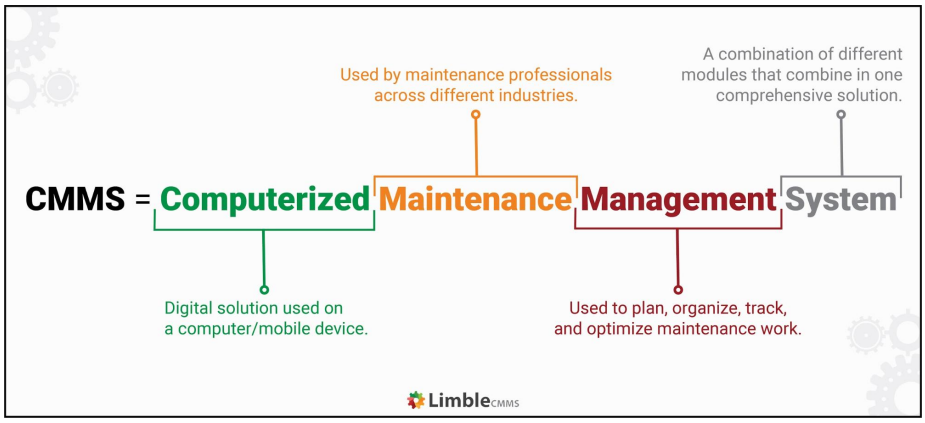

Figure 1: CMMS source(04/2024) : https://limblecmms.com/cmms/

Examples of CMMS

- **MaintainX** :

Figure 2: MAINTAINX source(04/2024) : https://www.prnewswire.com/news-releases/maintainx-raises-50m-with-12x-revenue-growth-from-boosting-productivity-of-industrial-and-frontline-workers-as-america-gets-back-to-work-301308326.html

MaintainX is a CMMS based on a mobile application and a web platform that aims to simplify the maintenance process for teams in the field.

It offers functionalities such as work order creation, asset management, spare parts inventory tracking, preventive maintenance task planning and report generation.

The mobile application allows technicians to receive real-time notifications, update the status of tasks and upload photos or notes to document interventions.

The web platform provides managers with a centralised dashboard where they can track the progress of jobs, analyse performance data and generate reports to optimise maintenance processes. source(04/2024)

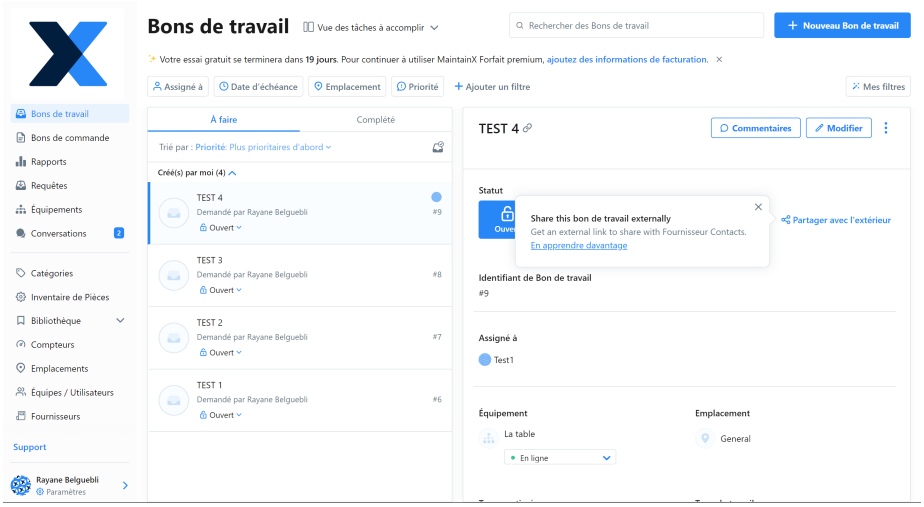

Figure 3: workorders

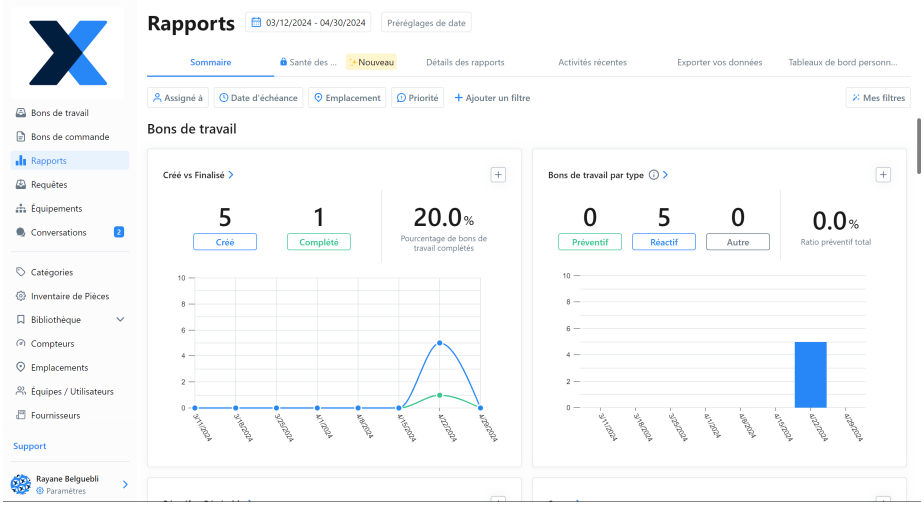

Figure 4: reports

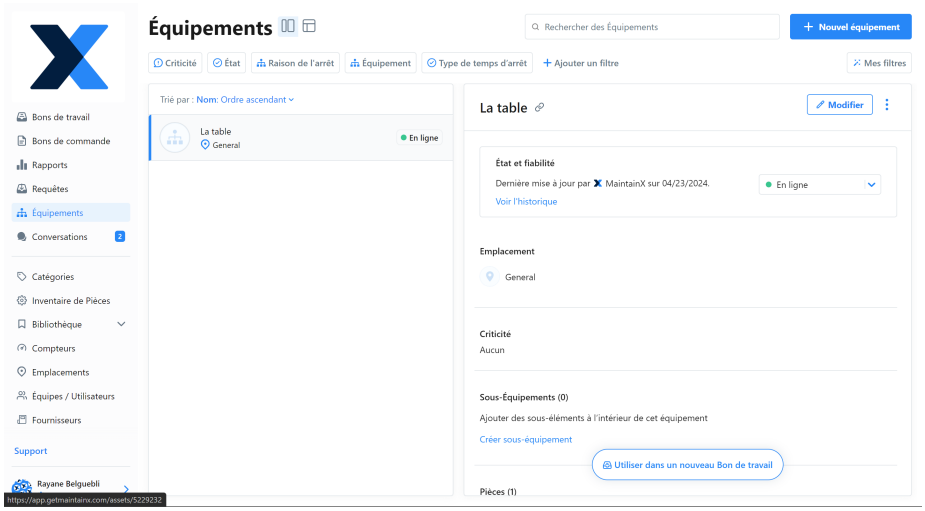

Figure 5: assets

- **Limble** :

Figure 6: LIMBLE source(04/2024) : https://www.prnewswire.com/news-releases/limble-announces-58m-series-b-funding-round-led-by-goldman-sachs-asset-management-bringing-total-valuation-to-450m-301857646.html

Limble is also a modern CMMS designed to simplify the management of maintenance activities and extend the life of assets.

It offers similar functionality to MaintainX, such as work order creation, scheduling, spares management and reporting.

Limble is distinguished by its user-friendly interface and advanced asset management features, such as equipment hierarchy modelling, warranty management and replacement planning.

It also incorporates predictive maintenance and performance dashboard capabilities to help users anticipate breakdowns and make informed decisions to optimise maintenance. source(04/2024)

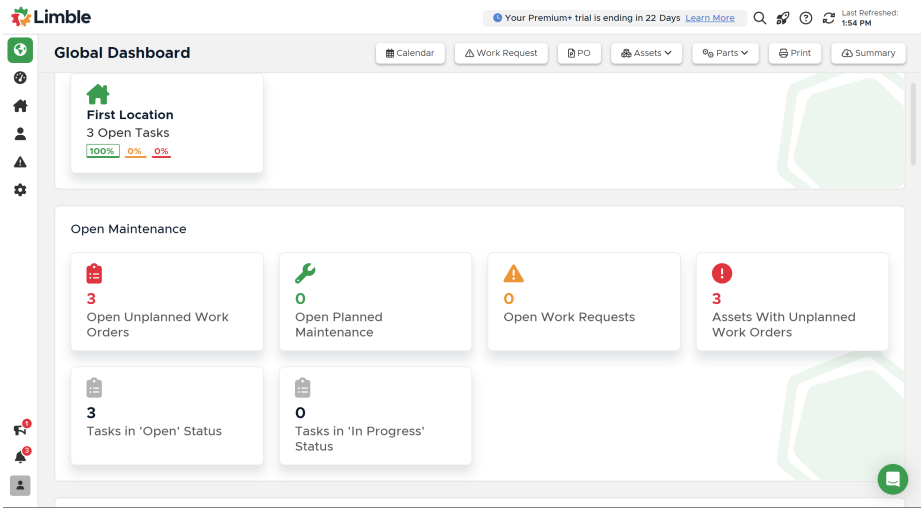

Figure 7: dashboard

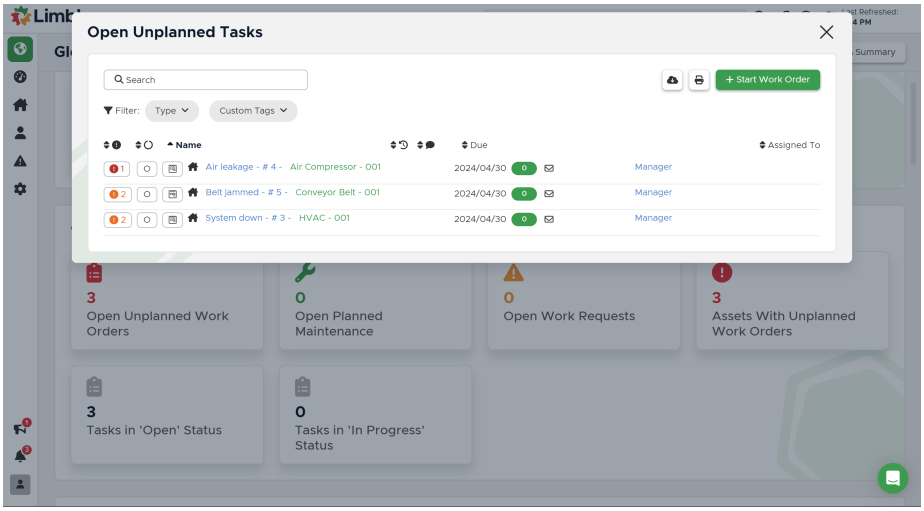

Figure 8: Tasks

- **Odoo Maintenance**

Figure 9: ODOO source(04/2024 : https://ubi.edu/odoo/)

Odoo is a suite of open source applications covering all your business needs: CRM, eCommerce, Accounting, Inventory, Point of Sale, Project Management, etc.

Odoo has a dedicated maintenance section that allows you to manage work orders and assets. Odoo Maintenance is perhaps less advanced than the previous CMMS but is sufficient to solve our problem and above all is free to download. source(04/2024)

Odoo is based on a 3-tier architecture:

- a PostgreSQL database server, which can contain several databases ;
- an application server containing the management objects, workflow engine, editing generator, etc. ;
- a presentation server that allows users to connect to OpenERP using any Web browser (with the Flash player installed for displaying graphics). This last server is not necessary if the user uses the native client, which does require installation on the user's workstation.

The server part is written in Python. The various building blocks are organised into modules. A module is a folder with a predefined structure containing Python code and XML files. A module defines the data structure, forms, reports, menus, procedures, workflow, etc. source(04/2024)

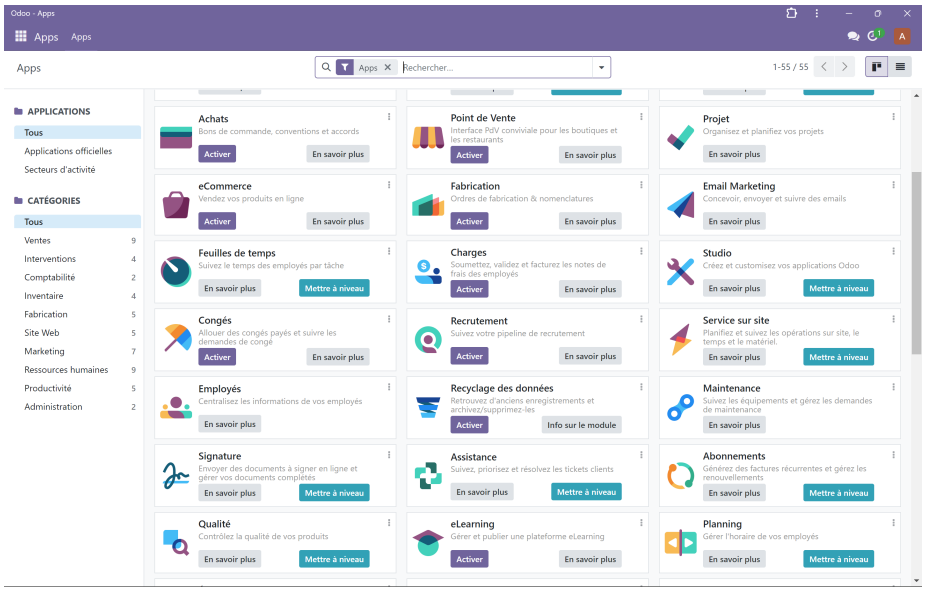

Figure 10: apps

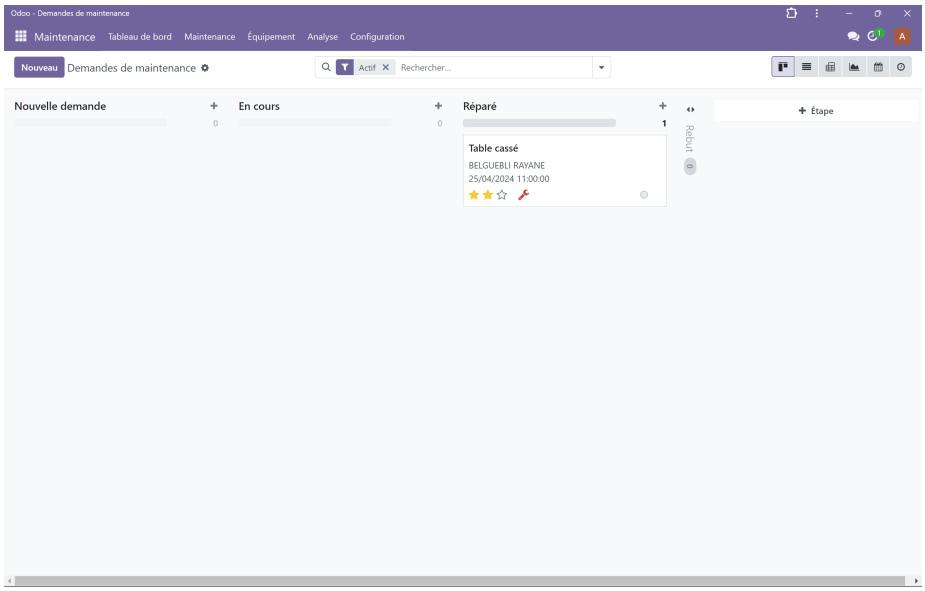

Figure 11: maintenace

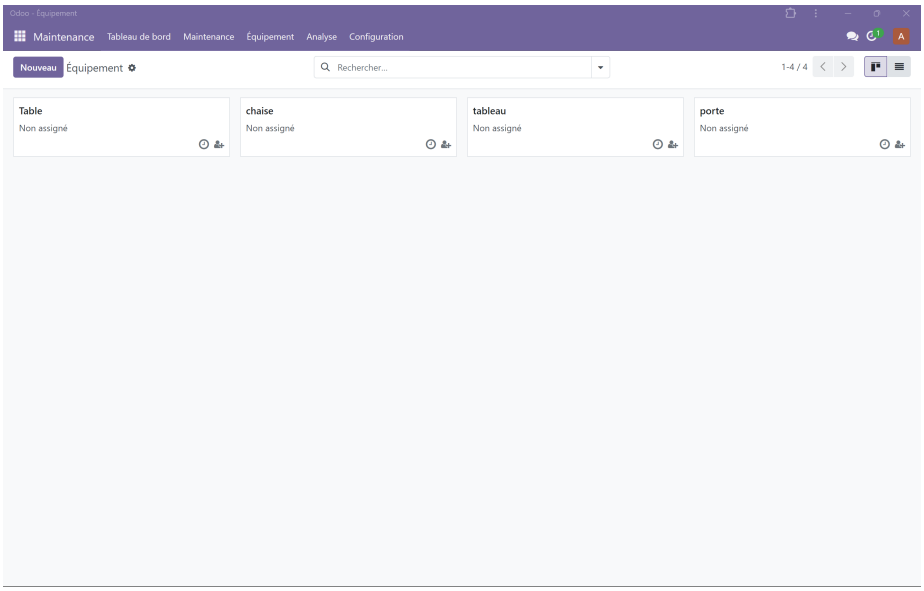

Figure 12: assets

CMMS comparison

	MaintainX	Limble	Odoo
Main features	work orders, digital checklists and real-time notifications.	work order, preventive maintenance, asset management and analytical dashboards.	work order management, preventive maintenance planning, asset tracking, spare parts stock management and report generation.
User interface	user-friendly mobile interface for fast, efficient communications.	intuitive interface, with simple navigation and clear menus.	user interface is renowned, ease of use and customisation.
Prices	paid enterprise version and online community version.	paid enterprise version and online community version.	paid enterprise version and open source community version.

	MaintainX	Limble	Odoo
Integration and customisation	MaintainX offers integrations with messaging tools and communication platforms such as Slack, as well as the ability to customise checklists and forms.	asset management systems, accounting software and other productivity tools.	other Odoo applications as well as with third-party software via additional modules.

What are Maintenance Management Systems?

Maintenance is a critical aspect of any manufacturing or industrial process. It ensures that equipment, machines and facilities operate at their optimum level, avoiding breakdowns and costly downtime. To achieve this, companies need to put in place a well-defined Maintenance Management System (MMS). An MMS is a holistic approach to maintenance management that involves the integration of people, processes and technology to optimise maintenance activities.

The main objective of an MMS is to improve equipment reliability, minimise downtime and increase productivity. It does this by providing a structured approach to maintenance planning, scheduling and execution. This helps maintenance teams to proactively identify and resolve potential problems before they become major issues. By leveraging data, analytics and technology, an MMS enables organisations to optimise maintenance processes, reduce costs and increase asset life. Here are some key points highlighting their importance and benefits :

- **Optimising maintenance operations** : MMS provides a centralised platform for planning, executing and monitoring maintenance activities. This enables efficient resource allocation, priority management and coordination of maintenance teams.
- **Reduced unplanned downtime** : By enabling preventive and predictive maintenance, MMS helps to identify and resolve problems before they become major breakdowns. This reduces unplanned downtime and minimises production interruptions.
- **Extended equipment life** : By providing regular, preventive maintenance, MMS helps to extend the life of equipment. By identifying and correcting potential problems at an early stage, MMS reduces wear and tear on assets.
- **Maximising asset availability** : By ensuring that equipment is well maintained and available when needed, MMS maximises asset availability. This enables organisations to maintain productivity and respond effectively to market demand.

In summary, MMS plays a crucial role in optimising maintenance operations, reducing downtime, extending equipment life and maximising asset availability. The above references illustrate the positive impact of MMS in various sectors and highlight their value as strategic tools for asset management and operational performance (Sorić 2024) [5].

What is the django framework et and why use this for our web application ?

Figure 13: DJANGO source(04/2024) : https://www.djangoproject.com/

In our project to create an innovative web platform that acts as a management centre for maintenance tasks, Django proved to be a wise choice for several key reasons :

- **MVC/MVT structure adapted to our architecture** : Django follows the MVT model, which is a variant of the MVC model. This architecture is particularly well-suited to our project because it allows us to clearly separate the different responsibilities of our application.
  - The Model represents data from the CMMS, such as maintenance task details, equipment information, schedules, etc.
  - The Template is responsible for presenting the data to technicians. In our case, this means generating content adapted to augmented reality for display on their glasses.
  - The View manages the business logic of our application, including the integration of data from the CMMS with augmented reality functionalities to provide technicians with precise, interactive instructions.
- **Managing complex data with Django's ORM** : Django's ORM (Object-Relational Mapping) is one of its most powerful and popular features. It greatly simplifies data manipulation by allowing developers to interact with the database using Python objects rather than direct SQL queries. Here are some key aspects of Django's ORM and why it's beneficial for our project:
  - **Database abstraction** : Django's ORM provides a high-level abstraction of the database, meaning developers don't need to worry

about the specific details of the underlying database (such as the type of database used or the SQL language).

This allows Django applications to be developed in a more portable way, as the code can be easily adapted to different types of database without having to change the business logic.

- **Using Python objects to represent data** : Database tables are represented by Python classes called "models" in Django. Each model defines the table's fields and relationships with other models.

  For example, in our project, we might have a Task model to represent the various maintenance tasks, with fields such as description, start date, end date, status, etc.

- **Easy data manipulation** : Once models are defined, CRUD (Create, Read, Update, Delete) operations can be performed using simple and intuitive methods on model objects.

  For example, to create a new task in our project, we would simply create an instance of the Task class, assign it the appropriate values for its fields, and then call the save() method to save it in the database.

- **Transparent management of relationships between tables** : Django's ORM makes it easy to define relationships between models, such as foreign keys and many-to-many relationships.

  For example, in our project, a task could be linked to a specific piece of equipment. This relationship can be easily defined in the Task model by adding a foreign key field that points to the Equipment model.

- **Protection against SQL injections** : Using Django's ORM provides built-in protection against SQL injections, as queries are securely generated using query parameters, preventing SQL injection attacks.

- **Flexible presentation of AR data** : Django has a flexible template system that allows dynamic content to be generated for a variety of devices, including augmented reality glasses.

  Templates can be customised to deliver instructions interactively, displaying details of maintenance tasks directly in the technicians' field of vision, improving their productivity and accuracy.

- **Security and scalability**: Security is a major concern, especially when it comes to handling sensitive data such as that associated with industrial maintenance. Django incorporates robust security features, such as protection against SQL injections and CSRF attacks, to ensure the security of user data.

  In addition, Django is known for its ability to manage large-scale web applications efficiently. Its scalability will enable our platform to grow

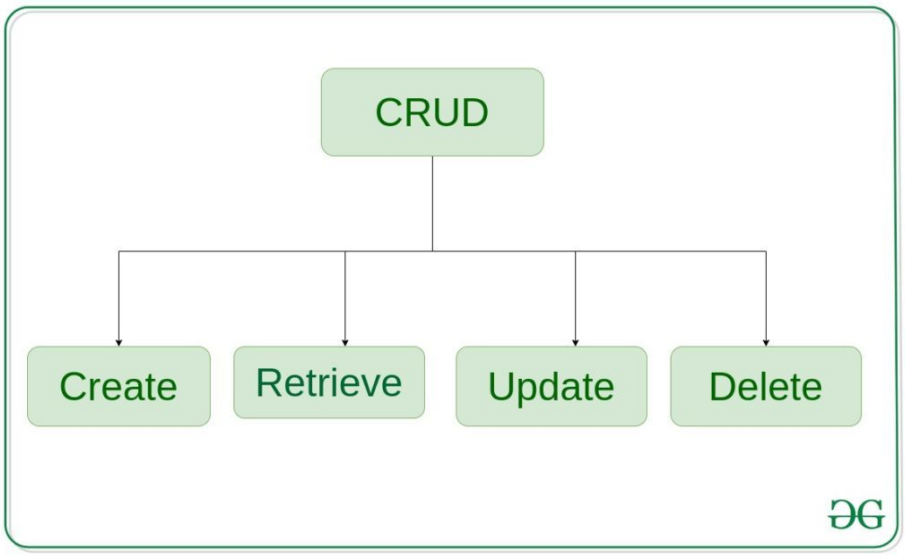

Figure 14: CRUD source(04/2024) : https://www.geeksforgeeks.org/django-crud-create-retrieve-update-delete-function-based-views/

with the addition of new features and larger volumes of data.

In short, Django offers a combination of powerful features, built-in security and flexibility that make it the ideal choice for our maintenance task management platform project incorporating augmented reality. Its MVC/MVT architecture, robust ORM and flexible templating system ensure that our solution can be implemented efficiently and scalably.

Requirements

Use Case

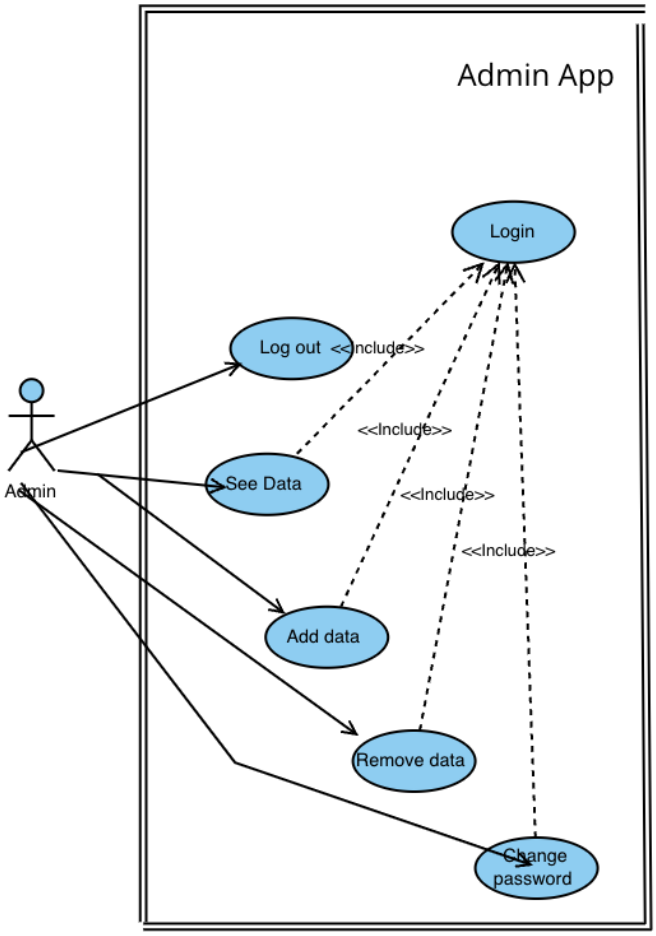

Figure 15: Use case for Admin

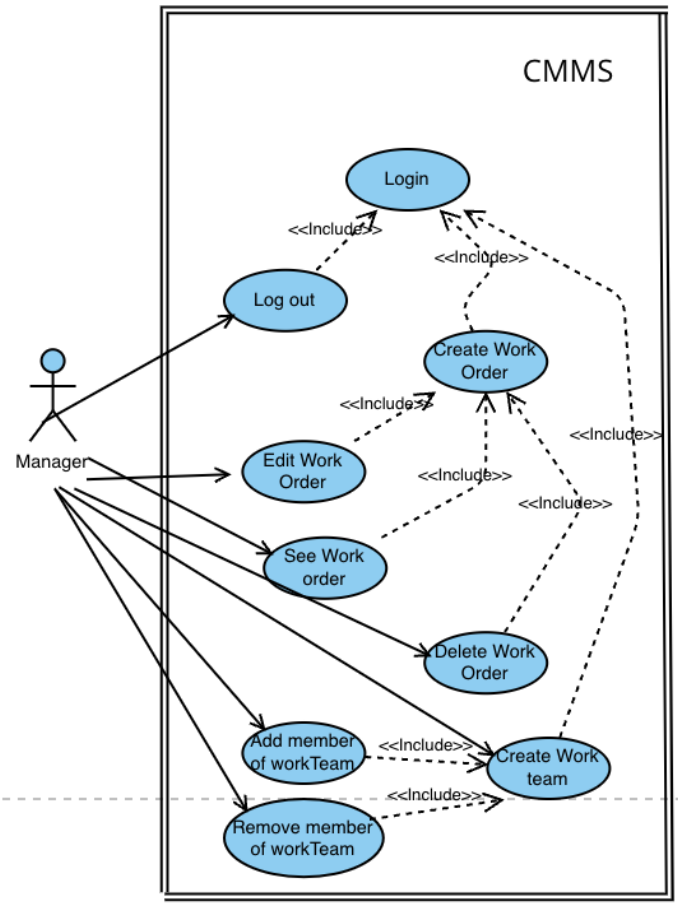

Figure 16: Use case for Manager

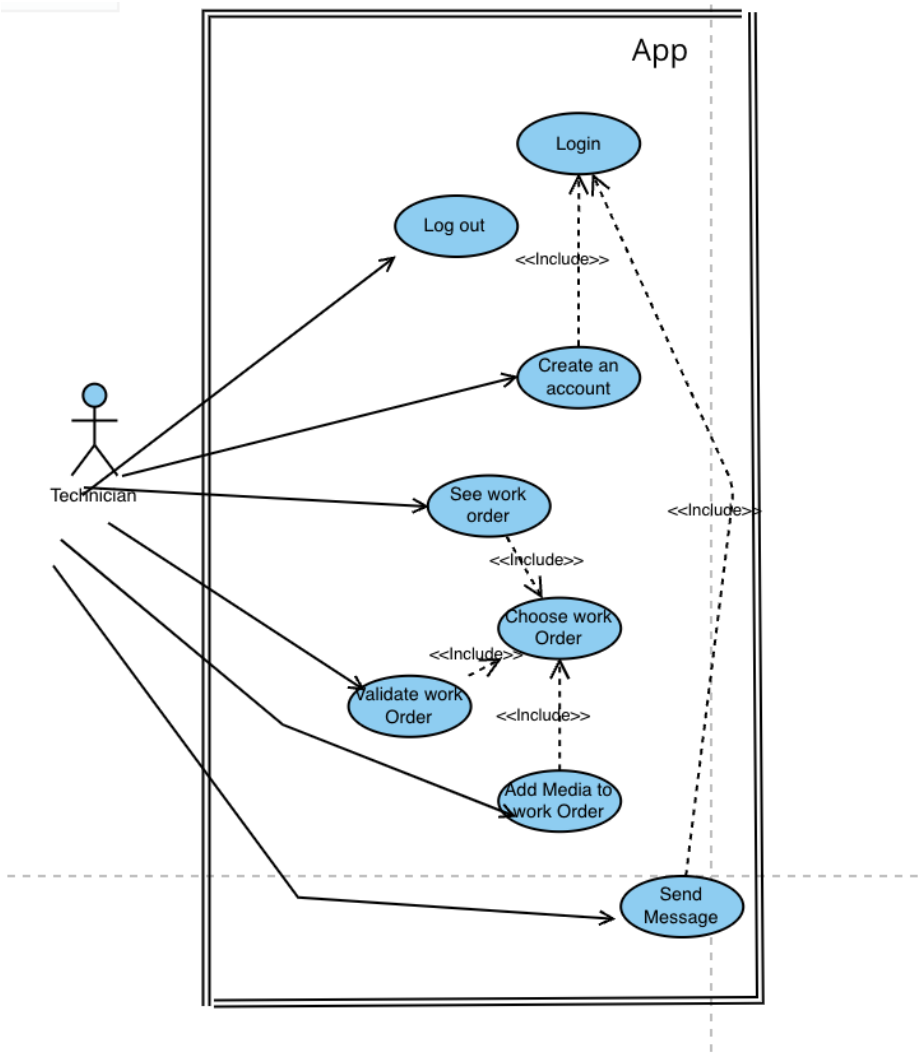

Figure 17: Use case for Technician

Descriptions

Admin App :

Login:

- **Primary Actor:** Admin
- **Objective:** To log in to access the features.
- **Preconditions:** The admin must be present on the site, connected to the Internet, and have an account.
- **Postconditions:** The admin is now logged into their account and can access the features.
- **Basic Flow:**
  1. The user clicks on the login button.
  2. A login form appears.
  3. The admin enters the required information.
  4. The admin clicks on the login button.
  5. A success message appears.
  6. The admin is now logged in and can access the features.
- **Alternative Paths:**
  - If the provided login information is incorrect, an error message appears, prompting the technician to check their information and try again.

Logout:

- **Primary Actor:** Admin
- **Objective:** To log out.
- **Preconditions:** The admin must be present on the site and logged into the platform.
- **Postconditions:** The admin is logged out of their account.
- **Basic Flow:**
  1. The admin clicks on the logout button.
  2. A logout confirmation message appears.
  3. The admin confirms the logout.
  4. A success message appears.
  5. The admin is now logged out.
- **Alternative Paths:**
  - If the admin simply closes the browser without logging out, an inactive session may automatically end after a certain period of time, with prior notification.

Add Data:

- **Primary Actor:** Admin
- **Objective:** Add data.
- **Preconditions:** The admin must be present on the site and logged into the platform.
- **Postconditions:** Data has been added.
- **Basic Flow:**

1. The admin clicks on the desired link at the top of the page (e.g., equipment, work orders,...).
2. A list of data and an "Add Data" button appear.
3. The admin clicks on the button.
4. A form is displayed.
5. The admin enters the desired information.
6. A success pop-up appears.

- **Alternative Paths:**
  - If the provided form information is incorrect, an error message appears, prompting the admin to check their information and try again.

Remove Data:

- **Primary Actor:** Admin
- **Objective:** Remove data.
- **Preconditions:** The admin must be present on the site and logged into the platform.
- **Postconditions:** Data has been removed.
- **Basic Flow:**
  1. The admin clicks on the desired link at the top of the page (e.g., equipment, work orders,...).
  2. A list of data and a "Remove Data" button appear for each item.
  3. The admin clicks on the button corresponding to the data they want to remove.
  4. A success pop-up appears.
- **Alternative Paths:**
  - If the data cannot be removed, an error message appears.

Change Password:

- **Primary Actor:** Admin
- **Objective:** Change password.
- **Preconditions:** The admin must be present on the site and logged into the platform.
- **Postconditions:** Password has been changed.
- **Basic Flow:**
  1. The admin clicks on the "My Account" button on the top menu of the page.
  2. A button labeled "Change Your Password" appears.
  3. The admin clicks on the button.
  4. A password change form is displayed.
  5. The admin enters their new password twice.
  6. They click on the "Change Password" button.
  7. A success pop-up appears.
- **Alternative Paths:**
  - If the admin enters a password that is the same as the previous one, an error message appears.

Main App :

Create an Account:

- **Primary Actor:** Technician
- **Objective:** To create an account to access the platform's features.
- **Stakeholders:** Social media company, social media users, subscribers, advertisers, investors.
- **Preconditions:** The technician must be present on the site and connected to the Internet.
- **Postconditions:** The technician now has an account and can access the features.
- **Basic Flow:**
  1. The user clicks on the create account button.
  2. A create account form appears.
  3. The technician fills in the mandatory information.
  4. The technician clicks on the create account button.
  5. A success message appears.
  6. The technician now has an account and can log in.
- **Alternative Paths:**
  - If mandatory information is missing, an error message appears, prompting the technician to complete the necessary fields.
  - In case of account creation failure due to technical issues, an error message appears, suggesting the technician to try again later

Login:

- **Primary Actor:** Technician
- **Objective:** To log in to access the features.
- **Preconditions:** The technician must be present on the site, connected to the Internet, and have an account.
- **Postconditions:** The technician is now logged into their account and can access the features.
- **Basic Flow:**
  1. The user clicks on the login button.
  2. A login form appears.
  3. The technician enters the required information.
  4. The technician clicks on the login button.
  5. A success message appears.
  6. The technician is now logged in and can access the features.
- **Alternative Paths:**
  - If the provided login information is incorrect, an error message appears, prompting the technician to check their information and try again.

Logout:

- **Primary Actor:** Technician
- **Objective:** To log out.

- **Preconditions:** The technician must be present on the site and logged into the platform.
- **Postconditions:** The technician is logged out of their account.
- **Basic Flow:**
  1. The user clicks on the logout button.
  2. A logout confirmation message appears.
  3. The technician confirms the logout.
  4. A success message appears.
  5. The technician is now logged out.
- **Alternative Paths:**
  - If the technician simply closes the browser without logging out, an inactive session may automatically end after a certain period of time, with prior notification.

View Work Orders:

- **Primary Actor:** Technician
- **Objective:** To view the list of work orders.
- **Preconditions:** The technician must be present on the site and logged into the platform.
- **Postconditions:** The technician views the available work orders.
- **Basic Flow:**
  1. The user clicks on a link in the top menu of the site.
  2. The list of work orders appears.
  3. The technician can now view the list of work orders.
- **Alternative Paths:**
  - If no work orders are available, a message indicating the absence of work orders appears, prompting the technician to try again later or contact support.

Select a Work Order:

- **Primary Actor:** Technician
- **Objective:** To choose a work order.
- **Preconditions:** The technician must be present on the site, logged into the platform, and on the work orders list page.
- **Postconditions:** The technician has now selected the desired work order, and that order no longer appears in the list.
- **Basic Flow:**
  1. The technician clicks on the desired work order.
  2. A “Choose this work order” button appears.
  3. The technician clicks on the button.
  4. The work order is now associated with the technician who chose it.
- **Alternative Paths:**
  - If the selected work order is already assigned to another technician, a warning message appears indicating that the work order is unavailable.

Validate a Work Order:

- **Primary Actor:** Technician
- **Objective:** To validate a work order.
- **Preconditions:** The technician must be present on the site, logged into the platform, and have already selected a work order.
- **Postconditions:** The technician has validated the work order, and it no longer appears in their list.
- **Basic Flow:**
  1. The user clicks on a link in the top menu of the site "View my work orders."
  2. The list of the technician's work orders appears.
  3. The technician clicks on the work order to be validated.
  4. A "Validate work order" button appears.
  5. The technician clicks on the button.
  6. A success message appears.

Add Media to a Work Order:

- **Primary Actor:** Technician
- **Objective:** To add media to a work order.
- **Preconditions:** The technician must be present on-site, logged into the platform, and have already selected a work order that is not yet validated.
- **Postconditions:** The technician has added media to a work order, and other individuals can view them.
- **Basic Flow:**
  1. The user clicks on a link in the site's top menu labeled "View my work orders."
  2. The list of the technician's work orders is displayed.
  3. The technician selects the desired work order from the list.
  4. An "Add Media" button appears.
  5. The technician clicks on the button.
  6. A media addition form appears.
  7. The technician uploads the desired media.
  8. The technician submits the form.
  9. A success message is displayed.
- **Alternative Paths:**
  - If the uploaded media is incorrect, an error message is displayed.

Send Message :

- **Primary Actor:** Technician
- **Objective:** To send message
- **Preconditions:** The technician must be present on-site, logged into the platform .
- **Postconditions:** Le technicien a envoyé un message a un autre employé .
- **Basic Flow:**
  1. The user clicks on a link in the site's top menu labeled "Chat"
  2. The list of the technicians displayed.
  3. The technician selects the desired technician from the list.

4. Un chat de message apparait .
5. Le technicien tape le message qu'il veut.
6. Le technicien appuie sur le boutton "envoyer".
7. Le technicien voit son message dans le chat .

Non-Functional Requirements

- Performance: The backend system should be highly performant and scalable to handle a large number of concurrent users and data requests. Optimize database queries and data processing to minimize response times and maintain overall application responsiveness.
- Security: Implement robust security measures to protect sensitive data, including encryption of data at rest and in transit.
- Maintainability: Employ modular and well-documented code to facilitate easy understanding, modification, and maintenance of the backend system.

References

Fu, Chuang, Lu-Qing Ye, Yuan-Chu Cheng, Yong-Qian Liu, and Benoi Iung. 2002. "MAS-Based Model of Intelligent Control-Maintenance-Management System (ICMMS) and Its Application" 1: 376–80.

Kour, Ravdeep, Miguel Castaño, Ramin Karim, Amit Patwardhan, Manish Kumar, and Rikard Granström. 2022. "A Human-Centric Model for Sustainable Asset Management in Railway: A Case Study." *Sustainability* 14 (2): 936.

Malta, Ana, Mateus Mendes, and Torres Farinha. 2021. "Augmented Reality Maintenance Assistant Using Yolov5." *Applied Sciences* 11 (11): 4758.

Shankar, Lakshmi, Chandan Deep Singh, and Ranjit Singh. 2021. "Impact of Implementation of CMMS for Enhancing the Performance of Manufacturing Industries." *International Journal of System Assurance Engineering and Management*, 1–22.

Sorić, Dario. 2024. "How to Optimize Operations and Maintenance for Success." *Maintenance World*.