الجمهـوريـــة الجزائـريـــة الديمقـراطيـــة الشـعبيـــة

République Algérienne Démocratique et Populaire

وزارة التـــعلـــيـــم العـــــالـــــي والبـحـــــث العـلمـــي

Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

جامعة وهران 2 محمد بن أحمد

Université d'Oran 2 Mohamed Ben Ahmed

معهد الصيانة و الأمن الصناعي

Institut de Maintenance et de Sécurité Industrielle

Département De Maintenance en Electromécanique

MÉMOIRE

Pour l'obtention du diplôme de Master

Filière : Electromécanique
Spécialité : Maintenance Fiabilité Qualité

Thème

Un Service Web Au Service De La Gestion De La Maintenance Assistée Par Ordinateur

Présenté et soutenu publiquement par :

Boudouaoui Taha et Riguet Hamza

Devant le jury composé de :

Nom et Prénom	Grade	Etablissement	Qualité
ADJELOUA Abd el Aziz	MAA	Université d'Oran 2	**Président**
ROUAN-SERIK Mehdi	MCB	Université d'Oran 2	**Encadreur**
REGUIEG Yssaad Sadek	MAA	Université d'Oran 2	**Examinateur**

Année 2020/2021

ملخص

الغرض الرئيسي من هذه الأطروحة هو تطوير نظام إدارة الصيانة المحوسب على شبكة الإنترنت .
الموقع الإلكتروني الذي قمنا بتطويره هو تطبيق ويب من صفحة واحدة يبسط للمجتمع الصناعي إدارة برامج الصيانة الخاصة بهم ، لمتابعة كل أداة في المخزون وجدولة الصيانة باستخدام أوامر العمل كما يسهل موقع الويب هذا مزامنة كل جهاز كمبيوتر في المنشأة الصناعية بحيث تتم مراقبة كل جهاز وتسجيل كل إجراء صيانة ليتم دراسته لاحقًا وتحسين التكاليف المادية والشخصية.

Résumé

L'objectif principal de cette thèse est de développer un système de gestion de maintenance informatisé basé sur le Web.

Le site Web que nous avons développé est une application Web d'une seule page qui simplifie pour la communauté industrielle la gestion de leurs programmes de maintenance, pour suivre chaque outil du stock et planifier la maintenance à l'aide de bons de travail.

De plus, ce site Web permet de synchroniser facilement tous les ordinateurs de l'installation industrielle afin que chaque équipement soit surveillé et que chaque acte de maintenance soit enregistré pour être étudié ultérieurement et optimiser les coûts matériels et personnels.

Abstract

The main purpose of this thesis is to develop a web-based computerized maintenance management system.
The website that we developed is a single page web application that simplify for the industrial community the management of their maintenance programs, to follow every piece of tool in the stock and schedule maintenance using work orders.
Also this website makes it easy to synchronise every pc in the industrial facility so that every equipment is monitored and every maintenance act is recorded to be studied later and optimize material and personal costs.

Acknowledgement

This project would not have been possible without the support of many people. Many thanks to our supervisor Mehdi Rouan Serik, who read our numerous revisions and helped make some sense of the confusion. Also thanks to our committee members, who offered guidance and support.

Thanks to the University of Oran 2 Mohamed Ben Ahmed and the Institute of Industrial Safety and Maintenance for awarding us all the means to complete this project.

Dedication

The sake of Allah, my Creator and my Master,
My great teacher and messenger, Mohammed (May Allah bless and grant him), who taught us the purpose of life,
I dedicate my dissertation work to my family, brothers and many friends. A special feeling of gratitude to my loving parents, whose words of encouragement and push for tenacity ring in my ears. My sister,have never left my side and is very special, and my cousin.
I also dedicate this dissertation to my many friends who have supported me throughout the process. I will always appreciate all they have done, especially Chaouki, Youssef, Hamoudi and my colleague Hamza.
I dedicate this work and give special thanks to my wonderful nieces Khalil and Amir.

Dedication

I give this dedication in the occasion of the completion of my master degree's thesis to every seeker for knowledge.
I dedicate my dissertation work to the one that stayed all night awake praying for me, and to my model in this life that taught me knowledge is a weapon in my life my preacious father, and to my sisters Amina, Hadjer, Hala and Asia, and to my brothers Talha and Abd El Kader.
I also dedicate my work to my lovely grandfather and grandmother, and to all my friends and lovers that supported me throughout my life, and many speacial thanks to my colleague and friend Taha, my friends Imad, Baset, Amine and Hamoudi.
I dedicate this work to all the teachers of the insitute.
And i finish my dedication with wishing by praying to our beloved prophet Mohamed (May Allah bless and grant him), and alhamdulillah for everything.

Contents

Chapter 1

General Introduction

Managing the maintenance of complex IT systems for organizations is not always easy. Sometimes, even simple tasks like file management can be difficult if someone else handles the task. Managing these tasks using web-based applications, or CMMS (customer relationship management software), can be quite useful in many ways. Here are some benefits to these types of programs.

Maintaining complex systems with web-based CMMS apps can provide significant cost savings by automating redundant tasks while reducing manpower requirements for data entry and manual updates.

Maintenance management is an important part of running a successful business. Whether it's an office, a residence, or another sort of property investment, if maintenance costs aren't kept track of on a regular basis, they can easily spiral out of control.

Proper maintenance management entails creating an annual budget tailored to your specific needs, determining how much work your facility need, updating the list of activities to guarantee they are accomplished, and assigning the right employees to each duty.

Maintenance management is not a chore that can be set and forgotten. As new solutions to solve difficulties emerge, there will always be a need for regular updating of knowledge.

Organizations now have the challenge of purchasing and administering various CMMS systems. There are many advantages to web-based CMMS systems, but the most important are the low costs of software licenses and support, the ease with which new updates or modules can be installed (thanks to centralized web update servers), the reduced need for on-site maintenance, the increased flexibility of user profiles (allowing users to work from anywhere), and the reduced hardware requirements.

There is now a cost-effective way to achieve all of the above with a single system. Other apps can be integrated into your CMMS using webservices, allowing them to send and receive data from your CMMS.

Chapter 2

General Information On CMMS

2.1 Introduction:

Maintenance management is a key component in the successful business operation. Whether it is an office, residential, or some other type of property investment where maintenance cost can sky-rocket quickly if not monitored on a regular basis.

Proper maintenance management includes developing an annual budget for your individual needs; assessing how much work your particular building needs; continually updating the list of tasks to ensure they are completed; and having the appropriate people assigned to each task.

Maintenance management does not come as a "set it and forget it" type of task. There will always be a need for constant updating of the information, as new ways to address issues arise. As technologies evolve, some tasks will be more quickly accomplished using some types of technology over others. In some cases, you may have to modify your maintenance plan, or use different equipment than you originally planned because of how the building itself is constructed.

When properly administered, maintenance management can result in less work, reduced costs for repairs and replacements; increased customer satisfaction; and put you back on track to profitability.

2.2 What is Maintenance Management? [18]

Maintenance management is defined as maintaining a company's assets and resources while controlling time and costs, thereby ensuring the maximum efficiency of the manufacturing process. Maintenance management has gone from an archaic, tedious, handwritten process to a computerized maintenance management system (CMMS), a software that plans, tracks, measures, and optimizes all forms of a maintenance program in one central system.

Maintenance management isn't just a software system. It's a combination of software, best practices, and trained personnel focused on the same goal. Maintenance management programs are highly customizable and centered around the type(s) of maintenance employed at a plant. Whether the company uses a condition-based maintenance program like predictive maintenance or a more time-based maintenance program like preventive maintenance, it's essential to focus its plan on the type of maintenance used and its role within your organization.

Improving maintenance management should be a continuous goal for any company with machine assets, but there is no one-size-fits-all solution. If the company needs additional support or expertise in shifting maintenance management practices, bringing in reputable reliability and maintenance experts could help it identify opportunities and create an achievable improvement plan.

2.2.1 Why Maintenance Management Is Important? [36]

Maintenance management is vital in ensuring the long-term success of maintenance programs by monitoring quality assurance, maintaining operational efficiency, and keeping assets in optimum running order. Properly maintained assets and resources keep the production stable and greatly minimize the chances for unplanned downtime. Unplanned downtime causes a snowball effect, leading to a spike in unexpected costs associated with things like repairs (overtime labor, spare parts, etc.), delayed shipments, lost revenue, or complete breakdowns of machines.

MAINTENANCE MANAGEMENT BENEFITS		
INDIRECT COSTS	DIRECT COSTS	CUSTOMER SERVICE
Minimize overtime	Reduce cost of maintenance systems by extending the useful life of assets	Improve reliability and availability
Minimize accidents	Reduce the cost of employing reactive maintenance personnel	Improve service performance
Minimize breakdowns	Reduce inventory costs	Reduce customer complaints
	Increase cost-effective decision making using historical data	Maximize product quality
		Maximize service delivery

Figure 2.1: Maintenance Management Benefits

Maintenance management helps improve the operational efficiency of plant facilities, which contributes to revenue by decreasing operating costs and improving the quality (and quantity) of manufactured products. In addition to cost savings, other benefits include improved workplace safety, enhanced productivity, and minimized human error.

2.2.2 Objectives of maintenance management: [19]

Almost any business process has objectives and maintenance management is no different. The eight main objectives are:

- Control Costs and Budget.
- Comply with Regulations.
- Plan Maintenance Work.
- Ensure Personnel Safety.
- Minimize Equipment Failure and Production Downtime.
- Extend Useful Machine Life.
- Improve Product Quality.
- Develop Improved Policies, Procedures, and Standards.

2.2.2.1 Control Costs and Budget:

The first objective is cost control and budgeting. The maintenance manager allocates the budget to various areas of the department's resources to ensure everything works effectively. It's crucial to include funds for both planned and emergency or corrective maintenance in the budget.

When budgeting for maintenance work, the important thing is that when a major asset breaks down completely, most likely, the asset will be replaced no matter the cost. Of course, a decision must be made between repairing the asset, replacing it, or purchasing a different type of machine. Maintenance managers are tasked with staying within the maintenance budget. However, when something like this happens, it justifies a maintenance budget overrun. They must make cost-effective, wise decisions when choosing parts, vendors, and new or replacement machines. They also need to balance the costs and benefits of preventive vs. corrective maintenance.

Labor resources are another area in which costs need to be monitored closely. When the budget allows, maintenance departments might outsource work to save time. However, this is not feasible in many cases, and corners get cut or tasks get delayed.

When there isn't enough time to complete all of the necessary corrective maintenance, there is no realistic way to implement a preventive maintenance program on top of what the team is already doing. Many organizations can implement this type of program only when they have the upfront resources and money available to invest in it. However, after they overcome that hurdle, the payback from this investment will come over time.

2.2.2.2 Comply with Regulations:

There are different types of regulations to follow, which vary by industry. Organizations in the oil and gas industry must follow FERC and EPA regulations, in addition to OSHA regulations. The food and beverage industry must follow many safety regulations, such as those advised by the FDA. Local, state, and federal regulation compliance standards must be considered by the production team at all times.

To maintain adherence to these regulations, it is essential to make small plans to accomplish bigger compliance goals. Examples of these plans include:

- Making sure there are adequate safety guards in place on all machines.
- Inspecting assets for damage.
- Having a system in place to report damage to assets and parts.
- Documenting and reporting on all repairs and replacements.

2.2.2.3 Plan Maintenance Work:

Strategic maintenance management includes scheduling maintenance jobs ahead of time. This is important because it efficiently distributes the appropriate time and labor resources to proactive, preventive maintenance tasks, helping maintenance departments reduce major asset failures. Maintenance managers must have a clear understanding of the structure of the company to schedule the work effectively. It will help determine the priority of various jobs.

For example, suppose a technician is installing new shelves in the stockroom and a pipe bursts in a bathroom. In that case, that is an emergency in which he may need to stop what's doing and help make that repair unless another technician is available. There are other scenarios in which the job prioritization is less obvious, but an experienced maintenance manager should help make those calls each day. The truck needs to be in service to make sure deliveries of that product are completed on schedule. If just a few tasks are prioritized incorrectly, the schedule can be thrown off for weeks. Maintenance work must be organized in a way that achieves optimal outcomes.

2.2.2.4 Ensure Personnel Safety:

Another objective of maintenance management is to ensure all personnel's safety, inside and outside of the maintenance department. This is done through regular inspections of boilers, compressors, material handling equipment, and other assets that could become dangerous if they malfunction. When maintenance is managed correctly, safety increases for everyone in the organization, many machines can be hazardous when operating normally, but even more so when it malfunctions. Proper training on the function of every asset, critical safety dos and don'ts, and emergency protocols are essential.

Facility management-related health and safety guidelines are also important. Just a few examples include:

- Sanitation and janitorial services.
- Pest control.

- Waste management.
- Managing HVAC, electrical, and plumbing work.

2.2.2.5 Minimize Equipment Failure and Production Downtime:

Maintenance teams strive to maximize equipment availability, and they are better able to do so when preventive maintenance jobs are managed well. Maintenance technicians must stay on top of preventive maintenance to keep machines running so that failure and production interruptions are minimized. At the same time, downtime cannot be avoided entirely, so when machines do need repairs, they must be done quickly and efficiently.

2.2.2.6 Extend Useful Machine Life:

When maintenance tasks are appropriately delegated, prioritized, and completed quickly, machines last longer. Over time a good maintenance plan improves reliability, availability, and maintainability. This is done through proactive maintenance work, which can include preventive, predictive, and condition-based maintenance. Preventive maintenance contains minor maintenance jobs and inspections to prevent asset breakdowns. Predictive maintenance uses real-time asset data collected through sensors, historical performance data, and advanced analytics to predict when a failure occurs. Condition-based maintenance uses real-time data to identify when an asset's performance or condition reaches an unsatisfactory level.

2.2.2.7 Improve Product Quality:

When machines are better maintained, the result is improved product quality and a decrease in the number of products that need to be scrapped or reworked. Improved product quality leads to better reviews of your product and more satisfied customers, and, in turn, more sales.

2.2.2.8 Develop Improved Policies, Procedures, and Standards:

A final maintenance management objective is to continually develop and improve upon policies, procedures, and standards that lead to better-maintained equipment and cost reduction. For this to be successful, there must be a mutual understanding between maintenance and other departments to plan, control, and direct maintenance activities. This includes knowing how to report maintenance issues to the maintenance team, what system to use to communicate the production and maintenance schedules, and how other groups will be notified of asset repair status.

2.2.3 Types of maintenance management [19]

Across the industry, many definitions are used when it comes to the different types of maintenance, the Figure 2.2 displays all the types.

2.2.3.1 Preventive maintenance (PM):

Preventive maintenance is a technique for maintaining equipment that involves replacing or restoring an asset at a predetermined interval, regardless of its condition. Preventive maintenance jobs include scheduled restoration and replacement tasks.

Preventive maintenance (also known as preventative maintenance) is a sort of maintenance performed on equipment at regular intervals. At the same time, it is still operational to prevent or minimize the likelihood of failure.

Preventive maintenance can be scheduled on a weekly, monthly, or three-monthly basis. However, preventive maintenance can also be based on usage, such as every 150 cycles, 10,000 hours, or 10,000 kilometers, like with a car.

Apart from the regular interval approach (time-based maintenance) there are also other types of maintenance that fall within the category of preventive maintenance:

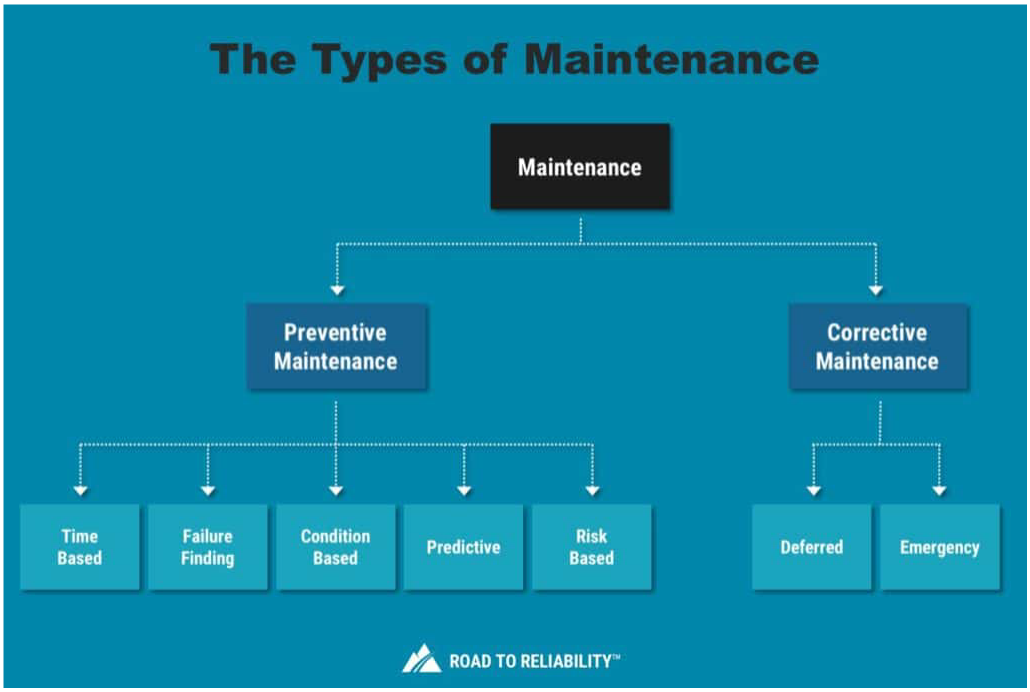

Figure 2.2: Types Of Maintenance

1. Time Based Maintenance (TBM).
2. Failure Finding Maintenance (FFM).
3. Risk Based Maintenance (RBM).
4. Condition Based Maintenance (CBM).
5. Predictive Maintenance (PDM).

Time-Based maintenance (TBM): Time-Based maintenance refers to replacing or renewing an item to restore its reliability at a fixed time, interval, or usage regardless of its condition; this is what "Moubray" calls Scheduled Restoration or Scheduled Discard tasks in his RCMII book.

That phrase is usually misunderstood as that other maintenance is not scheduled. When in fact, of course, all maintenance should be scheduled through a weekly schedule. The only exception would be Emergency Maintenance, which due to its very nature of requiring an immediate attention, cannot be scheduled.

Time-Based maintenance aims to protect the company against the failure of known wearing parts that have predictable Mean Time Between Failure (MTBF), i.e., Time Based Maintenance assumes that the failure is age-related and clear service life can be determined Or, that it is more economical and still (reasonably) practical.

Time-Based maintenance can never effectively manage non-age related failure modes, therefore, should only form a small part of the company's overall maintenance program as >70% of the failure modes in any plant are not age-related.

Failure Finding maintenance (FFM): Failure Finding Maintenance tasks are aimed at detecting hidden failures associated typically with protective functions. Think pressure safety valves, trip transmitters, and the like.

This type of equipment will not be required to function until something else has failed. That means that under normal operating conditions, one cannot know whether this equipment is still functional, i.e., the failure modes are hidden.

Moreover, since these failures are hidden, they need to find them before the company relies on that equipment to prevent an accident. Simple really. It is essential to realize that failure-finding maintenance tasks do not prevent failure but detect it, and once detected, it must be repaired. Failure Finding Maintenance is conducted at fixed time intervals typically derived from legislation or risk-based approaches.

Risk-Based maintenance (RBM): Perhaps the simplest type of maintenance management, (run to failure maintenance) is reaction-based. When an asset breaks, this maintenance is performed to get it back up and running again.

Condition-Based maintenance (CBM): Most failure modes are not age-related. However, most failure modes do give some warning that they are in the process of occurring or are about to occur.

If evidence can be found that something is in the early stages of failure, it may be possible to prevent it from failing entirely or from the consequences of failure. Condition Based Maintenance as a strategy looks for physical evidence that a failure is occurring or is about to occur. Thinking of CBM in this way shows its broader applications outside condition monitoring techniques often only associated with rotating equipment.

An important concept within Condition Based Maintenance is the P-F curve shown in the Figure 2.3: The curve shows that as a failure starts manifesting, the equipment deteriorates to the point

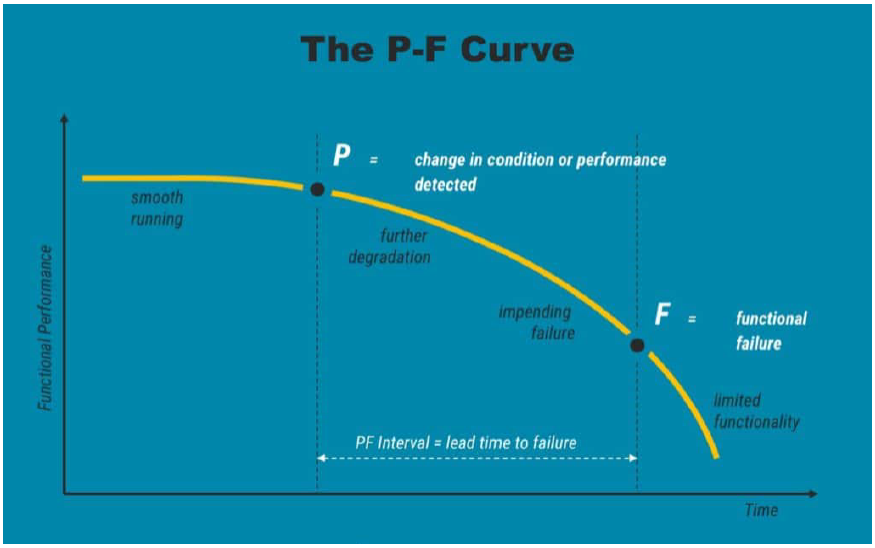

Figure 2.3: P-F Curve

at which it can be detected (point "P").

If the failure is not detected and mitigated, it continues until a functional failure occurs (point "F"). The time range between P and F, commonly called the P-F interval, is the window of opportunity during which an inspection can detect an imminent failure and give the company time to address it.

It is essential to realise that CBM as a maintenance strategy does not reduce the likelihood of a failure occurring through life-renewal, but instead is aimed at intervening before the failure occurs, on the premise that this is more economical and should have less impact on availability. In other

words: condition monitoring does not fix machines, and condition monitoring does not stop failures. Condition monitoring only lets the company find problems before they become a failure.

A common rule of thumb is that the interval between CBM tasks should be one-half or one-third of the P-F interval.

How much more effective CBM is above breakdown maintenance depends on how long the P-F interval is. The rectification can be planned with plenty of warning, materials and resources can be mobilised, and breakdown prevented (though production is still stopped for the maintenance duration). When the P-F interval is only a few days, the resulting organisational and workplace actions are much like a breakdown, and the value of CBM is primarily lost.

For CBM to be effective as a strategy, early intervention is essential. This requires an efficient and effective process for data gathering, data analysis, decision-making, and intervention.

For failure modes where the P-F interval shows a large variability, condition monitoring is not an effective strategy.

Predictive maintenance (PDM): Up until recently, when people spoke about Predictive Maintenance (PDM), this was usually a synonym for Condition Based Maintenance. But with the advent of Artificial Intelligence and the much lower costs of equipment sensors (IIoT – Industrial Internet of Things) and machine learning there is a difference appearing between Predictive Maintenance (PDM) and Condition Based Maintenance (CBM).

Predictive Maintenance is now considered an extension, a more advanced approach to CBM where we potentially use many process parameters gained from online sensors to determine if our equipment is moving away from stable operating conditions and is heading towards failure. A condition assessment that is extrapolated to make a prediction when failure is expected to occur.

Many (very large) companies are actively moving into this space, and it is certainly a fast-moving and exciting part of our discipline as Maintenance & Reliability specialized.

2.2.3.2 Corrective maintenance (CM):

A Run to Failure or Corrective Maintenance strategy only restores the function of an item after it has been failed. It is based on the assumption that the failure is acceptable (i.e., no significant impact on safety or the environment) and preventing failure is either not economical or not possible.

Apart from being the outcome of a deliberate Run to Failure strategy, Corrective maintenance is also the result of unplanned failures which were not avoided through preventive maintenance.

When opting for corrective maintenance as a strategy, it is essential to ensure that the failure modes under consideration do not have the potential to become Emergency Maintenance.

If a company adopt run-to-failure for equipment that once it has failed must be restored immediately, the company has doomed its self to a reactive maintenance environment.

A reactive maintenance environment is not where the company wants to be. It is more expensive, less efficient, and less safe.

Deferred Corrective maintenance: In the chart of maintenance types, the 'corrective maintenance' has been broken into two sub-types:

1. Deferred corrective maintenance
2. Emergency maintenance

Moreover, that was very deliberate because it is essential to minimise the amount of Emergency Maintenance allowed into the organisations.

As it has been pointed out above, Emergency Maintenance is expensive; various sources have suggested that Emergency Maintenance is 3 to 5 times as expensive as 'normal' preventive maintenance. Emergency Maintenance typically leads to more extended equipment outages, more production impact, and it is less safe.

So, when a corrective maintenance work request is raised, the company must prioritise it properly to defer the work request and give the associated team the time to plan and schedule the work properly.

Emergency maintenance (EM): Emergency Maintenance is corrective maintenance that is so urgent that it breaks into the company's Frozen Weekly Schedule. It upsets all plans and schedules and typically throws everything into disarray. Some people thrive in this environment and often get heralded as heroes when they have worked 16hrs non-stop to get production back online. However, when it comes to the Road to Reliability, it is a dead end.

Emergency Maintenance is the only maintenance type that the company must avoid as much as possible. World-Class organisations ensure that less than 2% of their total maintenance is Emergency Maintenance.

2.2.3.3 Types Of Maintenance (A Comparison Chart):

The Figure 2.4 shows a summary of:

- The different types of maintenance.
- What type of tasks are involved.
- The objective of the task.
- How the interval between the tasks is determined.

An efficient and effective Preventive Maintenance Program will have a mix of all these different types of maintenance.

Comparison of Maintenance Types

Maintenance Type	Preventive Maintenance					Corrective Maintenance	
	Time Based Maintenance	Failure Finding Maintenance	Condition Based Maintenance	Predictive Maintenance	Risk Based Maintenance	Deferred Maintenance	Emergency Maintenance
Task Type	Scheduled Overhaul / Replacement	Functional Test	Measurement of condition	Calculation and extrapolation of	Inspection or Test	Repair / Replace	Repair / Replace
Objective	Restore or replace regardless of condition	Determine if hidden failure has occurred	Restore or replace based on a measured condition compared to a defined standard		Determine condition and conduct risk assessment to determine when next inspection, test or intervention is required.	Restore or replace following failure. Result of a Run to Failure Strategy or an unplanned failure.	Restore or replace following unplanned failure.
Interval	Fixed time or usage interval e.g. 1 month, 1,000hrs or 10,000 km	Fixed time interval (can be set based on risk assessment e.g. SIL)	Fixed time interval for condition measurements / inspections		Time based interval between tasks and scope of task is based on risk assessment	Not applicable, but intervention is deferred to allow for proper planning & scheduling.	Immediate intervention required.

Figure 2.4: Comparison of Maintenance Types

2.2.4 Benefits of maintenance management:

Before we get into the specifics of what the benefits of maintenance management are, it's worth pointing out that many of these are positive results to the general objectives of maintenance management. That's not a coincidence; when the objectives are reached, these benefits flow over the entire company in question:

2.2.4.1 Cost savings:

Measuring and analysing assets on a regular basis also enables to see where improvements will need to be made. These costs can be calculated well in advance of the need, enabling to find the best price for new and/or improved assets.

2.2.4.2 Improve workplace safety:

Properly maintenance equipments and well-kept facilities prevent more employee accidents than most other business safety measures. This is particularly true on a day to day basis.

2.2.4.3 Enhance productivity:

When workplaces are safer, employees don't have to worry about as many daily risks, particularly risks due to equipments. This has a direct impact on overall productivity.

2.2.4.4 Minimizes human error:

A well-maintained facility also decreases the chances of human errors. This is particularly true when aspects of maintenance become automated.

2.2.4.5 Uncovers maintenance trends:

Finally, a quality maintenance management system uncovers what exactly is going on a day-to-day basis. If an asset, a piece of infrastructure, or other key part of a company's operation is consistently underperforming, one will find the fault much faster.

2.2.5 Conclusion:

What should we take away from this?

- Maintenance management is the process of maintaining the company's assets and resources.
- Its main objective is to streamline company processes and protect assets.
- Examples abound both of maintenance management done correctly and incorrectly.
- There are multiple types of maintenance management.
- Maintenance management solutions are worth the initial investment and time that they take to set up in order for company to reach its full potential.

2.3 What is CMMS? [9]

Computerized Maintenance Management System (CMMS), also known as **Computerized Maintenance Management Information System (CMMIS)**, is a software package that maintains a computer database of information about an organization's maintenance operations. This information is intended to help maintenance workers do their jobs more effectively (for example, determining which machines require maintenance and which storerooms contain the spare parts they need) and to help management make informed decisions (for example, calculating the cost of machine breakdown repair versus preventive maintenance for each machine, possibly leading to better allocation of resources).

CMMS data may also be used to verify regulatory compliance. To properly control the maintenance of a facility, information is required to analyse what is occurring. Manually, this requires a tremendous amount of effort and time. A CMMS also allows for record keeping, to track completed and assigned tasks in a timely and cost-effective manner. In recognition of this, companies have started using CMMS extensively to better control and organize maintenance management. The different steps of implementing a CMMS plan have been described in the diagram.

A CMMS offers multiple core maintenance functionalities. It is not limited to manufacturing but expands to facilities, utilities, fleet, hospitals, sports arenas and more where any type of equipment/assets are subject to repair and need maintenance. With improved technology and increasing competition, more and more companies are switching to CMMS vs using manual methods to track and organize information. The different components of a CMMS include but are not limited to:

1. Equipment data management.
2. Preventive maintenance.
3. Predictive maintenance.
4. Labour.
5. Work order system.
6. Scheduling/planning.
7. Vendor management.
8. Inventory control.
9. Purchasing.
10. Budgeting.
11. Asset tracking.

CMMS packages may be used by any organization that must perform maintenance on equipment, assets and property. Some CMMS products focus on particular industry sectors (e.g. the maintenance of vehicle fleets or health care facilities). Other products aim to be more general.

CMMS packages can produce status reports and documents giving details or summaries of maintenance activities. The more sophisticated the package, the more extensive analysis facilities have available.

Many CMMS packages can be either cloud-based, meaning they are hosted by the company selling the product on an outside server, or on-premises based, meaning that the company buying the software hosts the product on its own server.

2.4 CMMS application: [23]

The main advantage of CMMS is that it allows teams to track all their industrial maintenance operations or other activities in real time. Teams gain autonomy and use the tool to list, during the day, all the tasks they have been able to perform. Each operator, therefore, keeps track of his activity as well as the one of his colleagues' and access an agenda to plan his activities and to establish maintenance plans.

CMMS also allows each user to visualize all the breakdowns on the production line to keep track of all the defective equipment and of all the activities. Technicians and managers visualize them and become more aware of the several breakdowns occurring in their plant. Thanks to the maintenance management software, machine operators easily share information and gain time when they're on the field as they only work on the equipment that truly requires it.

More concretely, production profiles can enter all the information they think is necessary to allow teams in charge to perform 1st-level maintenance. As a consequence, the operator is more aware of his responsibilities and the team can focus more on critical equipment. The aim here is to enhance their reaction time.

Why centralizing information is crucial?

- Making the information easily accessible to all the company' services (maintenance, production, purchasing, store, accounting...) and allowing each user to have an overview of the data also helps to optimize the group's processes.
- To further share information, it is for instance possible to connect the maintenance management software to the company's ERP, in order to follow all the inventory levels as well as the spare parts. This data is essential as it allows a plant to manage its costs better. If maintenance teams know everything about the information related to the spare parts, purchasing

don't have to place huge and useless orders, which would otherwise lead to major money losses. There is indeed a strong need for transversality, and not only for technicians on the field.

- This principle is at the core of next-gen CMMS, which are radically different from more conventional tools, that became way too restrictive for companies willing to be always more innovative. Next-gen solutions even allow production operators to have access to all the maintenance-related activities of the company. Thanks to all the information given in by technicians, the tool becomes always more performing. In the equipment sheets available in the software, they can upload technical documents and other files, they register all the data about the machine (for example the date at which it was purchased) and access the whole intervention history.
- CMMS users can also find this information when they look at the catalogues of the official suppliers registered on the platform. Thanks to these products, users can get access to the suppliers' official documentation and can contact them if needed. They also have the possibility to copy these products in their equipment list without having to create them themselves as all the information is already provided by the supplier.

2.5 AI and CMMS: [27]

The last decade has seen maintenance management evolve immensely. From paper to spreadsheets to Computerized Maintenance Management Software (CMMS), it's been quite the journey for CMMS Software. The future of CMMS Software is predictive maintenance. However, with ever evolving technologies, what is it going to take to stay on top of maintenance management? We're hoping to provide some answers.

2.5.1 Introducing AI:

Did you know artificial intelligence was one of the most searched for keywords in Google this past year? Artificial intelligence is everywhere but do we understand what it really is? I'm assuming a bot comes to mind right away? Or a science fiction movie? Or battlebots?

Imagine you are entering your office facility, and the security system addresses you and opens the door. The security system quickly identifies you, does a background check from public and secure profiles and offers you a visitor's pass with your name written on it. Your office security systems are smart. Perhaps on their way of becoming intelligent. Artificial intelligence is a journey, with lots of auxiliary technologies involved. In this article we will try to explore the subject matter from maintenance management's perspective.

2.5.2 Pre-AI Maintenance Management:

Post industrial revolution, when we started getting a flavour of "bulk", (whether in terms of production, information or processing) we started organizing things on an industrial scale. Hence came the concept of standardization. And then a completely different era of evolution started. For example, repairing-jobs post a breakdown, evolved into preventive maintenance, paper and spreadsheets to track work order evolved to cloud based work order management.

With the onset of the ever increasing maintenance operations demands comes the need for a more evolved technology that addresses intelligent monitoring, prioritizing and optimizing maintenance schedules, self-maintenance and more!

2.5.3 AI in Maintenance Management:

A Boeing study suggests that 85% of equipment fails despite calendar-based maintenance and one-third of all maintenance investments are wasted through ineffective maintenance management methods. This need for a more thorough and accurate maintenance management software led to the birth of AI in maintenance management. Maintenance Management coupled with technologies like internet of things, big data and AI can revolutionize management. A study by Manufacturing

Business Technology stated that predictive maintenance using AI can save companies over 630$ billion in costs over the next 15 years.

All it needs is data. With each passing day, the rate at which data is captured and stored is increasing. A study from Gartner points out that 72 percent of manufacturing industry's data is unused due to the complexities involved with different variables, such as pressure, temperature and time. It's now getting humanly impossible to collate and process data using computers and processors. Augury, a New York-based predictive maintenance technology specialist, has its HVAC (heating, ventilation and air-conditioning) maintenance systems installed in more than 2,000 facilities across USA and Canada. Data gathered from one location goes into a "malfunction dictionary" that can be used for all.

This brings us to two important aspects of the use of AI in maintenance management:

2.5.3.1 Continuous-monitoring:

Continuous-monitoring as the name suggests is a vigilante system which involves both the failure system and the anomaly system. Failure system reads from the perspective of data patterns which indicate and predict operation failure. This means, like a doctor, the system knows what are the symptoms and indications of a failure.

On the other hand, an anomaly system reads data as deviations from normal routine operations. Unlike failure system, it does not focus on pre-defined failure symptoms but picks up variations from normal patterns. Both combined, give us a fair and a seamless monitoring of operational processes. Continuous-monitoring becomes important when low turn-around-time and down-time are of prime importance. It becomes even more important when operations involve very vital and critical processes. At times, they can be too complex and difficult for human interventions like gas pipelines over long distances, and tough climatic conditions.

2.5.3.2 Self-maintenance:

If maintenance is like friction, the necessary evil, self-maintenance is roller-skating. Someone wearing roller-skates is always in motion to avoid a fall. Self-maintenance combined with continuous-monitoring avoid the same thing, fall or breakdown. A breakdown or a maintenance task brings a set of activities with itself, like raising an alarm, initiating a work order, ordering a spare part etc. Many times the set of activities are pre-decided in form of an SOP, standard operating procedure. We rely on these SOPs as an alternative to human intelligence, mainly where a human intervention is not readily available.

2.5.3.3 Harnessing the AI wave: [15]

It's important to use the available technology to its utmost potential. Look for some relevant signs like developments in the areas of open data sources, machine learning and embedded AI. There are two categories of players who are active in these areas, the big enterprises like Amazon and Google, who've already accumulated massive amount of data. And the other, the relatively smaller players who act as solution providers or vendors. It is expected that these vendors would work on machine learning and open source (APIs and data) more rigorously, which they will eventually borrow from the big enterprises. These small players will develop tools which can be customized with the help of embedded AI and pre-built algorithms. Nobody must reinvent the wheel both in terms of data gathering and developing solutions. Existing codes and tools will be re-jigged and engineered to provide a custom fit solution. Of course this will give price benefits to clients and better margins to vendors as well.

Conclusively, it might be a good idea for CMMS Software service providers and vendors to start collecting relevant raw materials (codes and data sources), which can be quickly cooked into solutions, and thus served to their clients. For companies and solution providers, the best way to ride the AI wave is to find a right vendor who can bring technology and functional expertise on one platter, perhaps through using machine learning as a service. We know that technology plays an important role today in our day to day life. And it should in future. But the truth is, nobody knows

what exactly the future holds. At best, we can speculate with calculated risks, and not miss the first mover's advantage.

2.6 Conclusion

In this chapter we talked about maintenance management, its importance, its objectives and its types, and then we jumped to defining CMMS and its application, and then we have discussed the relationship between AI and CMMS. And in the next chapter we will go much more in depth talking about CMMS and especially the Web-Based CMMS.

Chapter 3

Web Services And CMMS

3.1 Introduction

Today, organizations are faced with issues of buying and managing multiple CMMS systems. There are many advantages to web-based CMMS systems, but most importantly the benefits include low cost for software licenses and support, less hassle in installing new updates or modules (with centralized web update servers), decreased need for on-site maintenance, increased flexibility with user profiles (allowing users to work from anywhere), and less hardware requirements.

These days there is a cost-effective way to have a single system do all of the above. Webservices allow other applications to be plugged into your CMMS so that they can send data to you as well as retrieve data from your CMMS.

3.2 Client-Server Architecture [5]

3.2.1 Definition[2]

A client-server architecture 3.1 divides an application into two parts, 'client' and 'server'. Such an application is implemented on a computer network, which connects the client to the server. The server part of that architecture provides the central functionality: i.e., any client can connect to the server and request that it performs a task. The server accepts these requests, performs the required task and returns any results to the client, as appropriate.

Consider an online bookstore as an example. The application allows a user to search and look at the details of an extensive range of books and then to order a book. The application software provides an interface and a means of selecting or finding a book's details and displaying book information, and allowing a book order to be generated.

The application could take the form of a single 'chunk' of software downloaded from the web. However, if the software is one monolithic item, every time anything is changed or updated, the entire application has to be redistributed again. This would not work well in this example because the catalogue of books will change regularly. An improvement might be to split the application into two parts. One, the client, can provide the interface for users and be distributed to them. The other part can be kept and run on the company's own server machine 3.1. The client application can display information and pass information to the server for searching, such as the title of a book. This client application, or software, is quite commonly called the 'presentation' layer or tier.

Figure 3.1: Client-Server Architecture

In this client-server model, many clients can connect to the server application and request information about books. The server has to process these requests and send the response to the client that originated the request and not to any other client. As long as the network is working well and the server can respond to all the requests it receives, such a 'split' application will provide much

the same level of service as the monolithic version. This simple client-server architecture is also commonly called 'two-tier architecture'.

The catalogue of book information can be held centrally on the server and then be easily updated. This allows other 'centralised' data to be maintained and sent to clients, such as the stock level of each book. Users of the client will find it much more straightforward and smaller to work with than the complete application. At the same time, the company will have better control and be able to, for example, monitor usage of the server application itself. A standard client used to access applications is a web browser that accesses server applications (such as applications on websites) using HTTP. Using a web browser as the client end of an application is interesting because, for most applications, the browser is provided by a third party. This means that application builders must rely on agreed standards for the behaviour of the client component.

There are also different distributions of functionality across a two-tier architecture. For instance, suppose you have a client that accesses your bank account online. For example, if that client is a web browser, it can be used to request and display your accounts' statements from the bank's server. The information you may obtain is restricted to the pre-defined views of the information provided by the server. So, while the server's information might include your account balance, if you want to find out the total payments in and out over the last week or year, you will still have to calculate it yourself, based on the figures the server provides. Alternatively, you might access your bank server over the internet using another, 'more intelligent' client, such as a mobile app. This client might include a facility to extract figures from your bank statements and perform whatever calculations you require. Such a client might also create bar or pie charts that display your income and expenditure across different categories that you define.

The web browser client in the example of the online bank displays the information that the server provides. The 'more intelligent' mobile app client allows you to take the server's information and manipulate and display it in various ways according to your personal needs. The web browser with little functionality is often termed a thin client, while the more intelligent client is usually a thick (or 'thicker') client.

We have seen that a two-tier approach seems to have some advantages, at least for applications that operate over networks. The client that is distributed to users may change. At the same time, the server part can be a centralised component that maintains dynamic, global data consistently and securely for the organisation and for users to access and use. Suppose a third-party component, such as a web browser, provides the functionality required to support the application. It can be adopted as part of the solution, with a significant saving of development effort. There is a potential disadvantage to splitting the application across a network because that data has to be transmitted over a slow or unreliable connection.

There are other, slightly less apparent advantages in breaking the application into components in this way. We shall look at some of these in more detail shortly, but let's look at one advantage now. This is an advantage you will have benefited from if you have ever decided to change your web browser. For example, you might have changed from Internet Explorer to Mozilla Firefox, just as a personal preference. When you did this, you changed the client component of all the online applications you use. This is only possible because web browsers are mainly based on common standards. They are not realised as an intrinsic or built-in part of any of the applications you use; in other words, the client is loosely coupled to the server application.

3.2.2 Types of Client-Server Architecture [4]

3.2.2.1 One-Tier Architecture

In the One-Tier architecture, all client/server configuration settings, user interface environment, data logic, and marketing logic system exist on the same system. These services are reliable, but it isn't easy to handle them because they contain all data in different variances, which are allotted the replication of the entire work. This architecture also contains different layers.

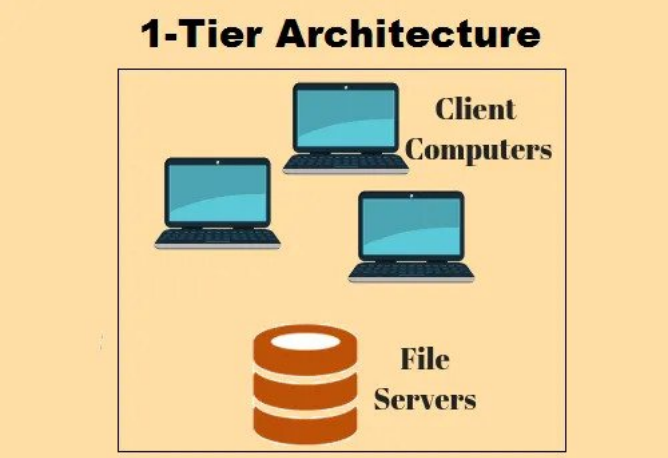

Figure 3.2: One-Tier Architecture

3.2.2.2 Two-Tier Architecture

Two-Tier architecture provides the best client/server environment that helps store user interface on the client system, and all database is saved on the server machine. Business logic and database logic exist on the client otherwise server, but they must be maintained. When data logic and business are gathered on the client terminal, it is known as "fat client thin server architecture". But if Business Logic and Data Logic are controlled at the server machine, it is known as "thin client fat server architecture".

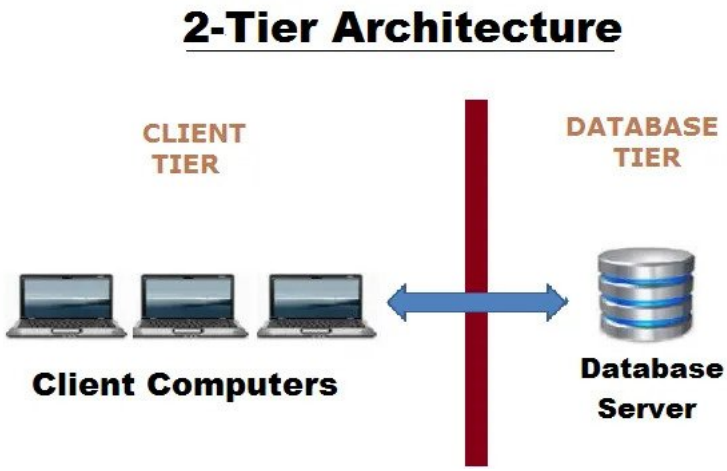

Figure 3.3: Two-Tier Architecture

In this architecture, client and server machines are connected directly because if the client is firing any input for the server terminal, then in between should not be any intermediate. So, it delivers the output with the fastest rate and to ignore misunderstanding between the other clients.

Its benefits are:

- Easy to design all applications
- Maximum user satisfaction

- Implementation of Homogeneous Environment
- Best performance

Its Limitations are:

- Poor performance due to grow number of connections of each user.
- Less security.
- All clients are totally dependent upon the manufacturer's database.
- Less portability means this architecture is totally dependent upon the particular database.

3.2.2.3 Three-Tier Architecture

In this Three-Tier architecture, middleware is needed because if the client machine sends the request to the server machine, the middle layer receives this request, and finally, this request is sent to the server. So, firstly, the server's response is received by the middle layer and sent to the client machine. All data logic and business logic are stored on the middleware. Due to the use of middleware, flexibility is greatly improved, and excellent performance is delivered.

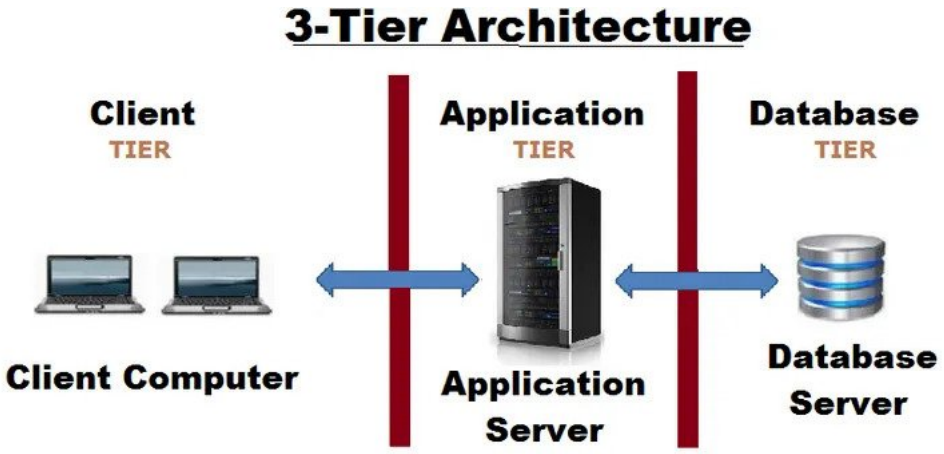

Figure 3.4: Three-Tier Architecture

Three-Tier architecture is divided into three layers, the presentation layer (Client Tier), the Application layer (Business Tier) and the Database layer (Data Tier). The client machine handles the presentation layer, the Application layer controls the Application layer, and finally, the Server machine takes care of the Database layer.

Its benefits are:

- Best performed data integrity.
- Improved security to Two-Tier architecture.
- Hide database structure.

Its limitations are:

- To increase complexity of communication in between client and server because in which middleware is also used.

3.2.2.4 N-Tier Architecture

This architecture is also known as the "Multitier Architecture", so it is a scaled form of Three-tier architecture. In this architecture, entire presentations, application processing, and data management functions are isolated.

Its benefits are:

- It delivers the flexible and reusable applications.

Its limitations are:

- Harder to implement because it uses the complex structure (componentization of tiers)

3.2.3 Examples of Client Server Architecture [4]

There are four examples of Client-Server Architecture:

3.2.3.1 Web Servers:

Web server likes as a high-performance computer system that can host multiples websites. This server installs different types of web server software like Apache or Microsoft IIS, which delivers access to hosted several websites on the internet. These servers are linked with the internet through a high-speed connection that provides ultra data transmission rates.

3.2.3.2 Mail Servers:

Email servers help to send and receive all emails. Some software is run on the mail server, which allows the administrator to create and handle all email accounts for any domain hosted on the server. Mail servers use some protocols for sending and receiving emails, such as SMTP, IMAP, and POP3. SMTP protocol helps to fire messages and manages all outgoing email requests. IMAP and POP3 help to receive all messages and handle all incoming mails.

3.2.3.3 File Servers:

The file server is a dedicated system that allows users to access all files. It works as a centralized file storage location, and several terminal systems can access it.

3.2.3.4 DNS:

DNS stands for "Domain Name Server", and it has a massive database of different types of public IP addresses, and they link with their hostnames.

These server types help deliver all resources (like files, directories, shared devices such as applications and printers) to client terminals like PCs, smartphones, PDAs, laptops, tablets, etc.

3.2.4 Components of Client Server Architecture

The client-server architecture contains three components: workstations, server, and networking devices, and they are connected, and bellow we go into some details:

3.2.4.1 Workstation:

Workstation is also known as "Client Computer". There are different operating systems installed on the workstations like Windows 2000, Windows XP, Windows Vista, Windows 7, and Windows 10. These workstation operating systems are cheaper compare to the server's operating systems.

3.2.4.2 Server:

The server is an ultra-performance computer system that contains a fast memory, ample hard drive space, and high-speed processors because they save and service several requests from the workstation side. A server plays different types of roles like a mail server, a database server, a file server, and a domain controller simultaneously.

3.2.4.3 Network Devices:

With the help of network devices, workstations and servers are connected. Every network device has its functionality. A hub connects the server to multiple workstations, a repeater is used for moving data from one device to another, and bridges help isolate all network segments.

3.3 Websites [31]

3.3.1 Definition

A website is a collection of HTML documents that can be called up as individual web pages via one URL with a client such as a browser.

Alongside classic content such as text, internet sites can also include image media like photos or videos and other files. Furthermore, web applications offer the opportunity for users to interact with a website. The subpages of a website are correctly described as web pages. Alternative terms include homepage or internet site.

3.3.2 The creation of websites [6]

In 1990, the first website in the world was published by an employee of the CERN research project in Geneva. Just three years later, the World Wide Web began with websites that could be called up all over the world via an internet connection.

The first websites were made up almost exclusively of text and a few pictures. Throughout the commercialization of the internet and the expansion of bandwidth by internet providers, websites have become more and more complex. Thanks to the increasing mobile use of the internet, however, a trend for simplification in design has resurfaced.

Today there are more than 1.3 billion websites worldwide, from small sites with a single URL, as can be created with a homepage construction kit, to complex, professionally created online shops with many thousands of subpages. Interestingly, a large part of internet sites used worldwide cannot be accessed via public networks. In some cases, these are internal company intranets, while in other cases, there is "dark web" content, which can only be accessed with specific browsers. The actual number of websites could therefore have long since broken the barrier of 1.3 billion mentioned above.

The global traffic related to websites creates a daily data quantity of more than four billion gigabytes.

3.3.3 Structure of a website

Nowadays, a website is mainly made up of numerous web pages. These web pages are HTML documents that are stored in a directory on a domain. The domain should not be confused with the website. The domain is the internet address via which the content of the website can be called up.

HTML is not a programming language but a page description language. It defines which element is to be displayed at which point. This means in the design of a website, there is no picture used for the entire site. Instead, it is defined in the code how the font, colors, and possible images or videos are to be arranged – depending on what device is being used (e.g., desktop computer or smartphone) to access it. When a website is designed to adapt to mobile requirements, it's called responsive design.

The individual webpages of a website are stored in directories, which each form their URL. If the user enters this URL, they can call up the desired website. The search engine also holds the URL

for the individual websites in the so-called "search snippet" with elements such as meta description, meta title, or rich snippets displayed in the search results as a clickable link.

3.3.4 Creating websites

A homepage can be created with the help of a simple text editor. This allows users to add HTML elements. This HTML file is then uploaded to a directory or a domain on a server. In this way, a client can call up the website with its content via the web.

HTML documents can include paths for different file types, such as CSS files or JavaScript files. They increase functionality or can influence the layout of a website.

Many webmasters do not have an individual website programmed but either uses a "homepage construction kit" or so-called "CMS,," i.e., content management systems. These are programs that companies use to and professionally create websites. Popular CMS providers include WordPress, Joomla!, and Wix.

3.3.5 Types of websites

There are a wide variety of website types that can be found on the World Wide Web, and we state some of them below:

3.3.5.1 Blogs:

These are the most common form of websites on the net. Anyone who wants to create a website nowadays mainly uses the form of a blog. A blog in its original form is characterized by a private user writing about issues from their subjective point of view and publishing to their blog. Today blogs are used for a variety of different purposes. For example, companies often have a so-called "corporate blog," Employees write about company-relevant topics.

3.3.5.2 Online Shops:

A webshop is a website designed for online shopping. The site offers information on products or services and allows customers to purchase or order these items online.

3.3.5.3 Web Directories:

In the early days of the commercial internet, search engines were not as efficient as they are now. Users, therefore, often used so-called "web directories" to find internet sites on a specific topic. Nowadays, web directories play a much smaller role, as they are also often used for search engine spam.

3.3.5.4 Price Comparison Portals:

These websites offer buyers the opportunity to compare prices for a particular product.

3.3.5.5 Forums:

In forums members, can log in and discuss their own chosen topics online.

3.3.6 News Websites:

These are generally digital forms of classic news magazines. Today the digital versions have significantly higher coverage than the print editions.

3.3.6.1 Social Networks:

Social networks such as Facebook or Twitter are unique websites that allow users to interact. You do not need your domain to use these portals.

3.3.6.2 Web Apps:

For example, Web apps used on smartphones or computers are not amongst the classic websites. They are comparable with software that is installed on a computer. Web apps differentiate themselves by generally only being usable with an internet connection. Web apps for Android devices can be given their own URIs as part of app indexing by Google. In this form, web apps are, in turn, comparable with classic websites. However, they have been designed exclusively for mobile end devices and not for desktop PCs.

3.4 Database [32]

3.4.1 Definition

A database is an organized collection of structured information, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the associated applications, are referred to as a database system, often shortened to the database.

Data within the most common types of databases in operation today is typically modelled in rows and columns in a series of tables to make processing and data querying efficient. The data can then be easily accessed, managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data.

3.4.2 Evolution of the database

Databases have evolved dramatically since their inception in the early 1960s. Navigational databases such as the hierarchical database (which relied on a tree-like model and allowed only a one-to-many relationship), and the network database (a more flexible model that allowed multiple relationships), were the original systems used to store and manipulate data. Although simple, these early systems were inflexible. In the 1980s, relational databases became popular, followed by object-oriented databases in the 1990s. More recently, NoSQL databases came about as a response to the growth of the internet and the need for faster speed and processing of unstructured data. Today, cloud databases and self-driving databases are breaking new ground regarding how data is collected, stored, managed, and utilized.

3.4.3 The difference between a database and a spreadsheet

Databases and spreadsheets (such as Microsoft Excel) are both convenient ways to store information. The primary differences between the two are:

- How the data is stored and manipulated
- Who can access the data
- How much data can be stored

Spreadsheets were initially designed for one user, and their characteristics reflect that. They are great for a single user or a small number of users who do not need to do a lot of incredibly complicated data manipulation. On the other hand, Databases are designed to hold much more extensive collections of organized information in massive amounts, sometimes. Databases allow multiple users at the same time to quickly and securely access and query the data using highly complex logic and language.

3.4.4 Types of databases [21]

There are many different types of databases. The best database for a specific organization depends on how the organization intends to use the data.

3.4.4.1 Relational databases:

Relational databases became dominant in the 1980s. Items in a relational database are organized as a set of tables with columns and rows. Relational database technology provides the most efficient and flexible way to access structured information.

3.4.4.2 Object-oriented databases:

Information in an object-oriented database is represented in the form of objects, as in object-oriented programming.

3.4.4.3 Distributed databases:

A distributed database consists of two or more files located in different sites. The database may be stored on multiple computers, located in the exact physical location, or scattered over different networks.

3.4.4.4 Data warehouses:

A central repository for data, a data warehouse is a type of database designed explicitly for fast query and analysis.

3.4.4.5 NoSQL databases:

A NoSQL, or nonrelational database, allows unstructured and semistructured data to be stored and manipulated (in contrast to a relational database, which defines how all data inserted into the database must be composed). NoSQL databases grew popular as web applications became more common and more complex.

3.4.4.6 Graph databases:

- A graph database stores data in terms of entities and the relationships between entities.
- OLTP databases. An OLTP database is a speedy, analytic database designed for large numbers of transactions performed by multiple users.

These are only a few of the several dozen types of databases in use today. Other, less common databases are tailored to particular scientific, financial, or other functions. In addition to the different database types, changes in technology development approaches and dramatic advances such as the cloud and automation are propelling databases in entirely new directions. Some of the latest databases include:

3.4.4.7 Open-source databases:

An open-source database system is one whose source code is open-source; such databases could be SQL or NoSQL databases.

3.4.4.8 Cloud databases:

A cloud database is a collection of structured or unstructured data that resides on a private, public, or hybrid cloud computing platform. There are two types of cloud database models: traditional and database as a service (DBaaS). With DBaaS, administrative tasks and maintenance are performed by a service provider.

3.4.4.9 Multimodel database:

Multimodel databases combine different types of database models into a single, integrated back end. This means they can accommodate various data types.

3.4.4.10 Document/JSON database:

Designed for storing, retrieving, and managing document-oriented information, document databases are a modern way to store data in JSON format rather than rows and columns.

3.4.4.11 Self-driving databases:

The newest and most groundbreaking type of database, self-driving databases (also known as autonomous databases), are cloud-based and use machine learning to automate database tuning, security, backups, updates, and other routine management tasks traditionally performed by database administrators.

3.4.5 What is database software?

Database software is used to create, edit, and maintain database files and records, enabling more accessible file and record creation, data entry, data editing, updating, and reporting. The software also handles data storage, backup and reporting, multi-access control, and security. Strong database security is fundamental today as data theft became more frequent. Database software is sometimes also referred to as a "database management system" (DBMS).

Database software makes data management simpler by enabling users to store data in a structured form and then access it. It typically has a graphical interface to help create and manage the data, and, in some cases, users can construct their databases using database software.

3.4.6 What is a database management system (DBMS)?

A database typically requires a comprehensive database software program known as a database management system (DBMS). A DBMS serves as an interface between the database and its end-users or programs, allowing users to retrieve, update, and manage how the information is organized and optimized. A DBMS also facilitates oversight and control of databases, enabling various administrative operations such as performance monitoring, tuning, and backup and recovery.

Some examples of popular database software or DBMSs include MySQL, Microsoft Access, Microsoft SQL Server, FileMaker Pro, Oracle Database, and dBASE.

3.4.7 Using databases to improve business performance and decision-making

With massive data collection from the Internet of Things transforming life and industry across the globe, businesses today have access to more data than ever before. Forward-thinking organizations can now use databases to go beyond basic data storage and transactions to analyze vast quantities of data from multiple systems. Using database and other computing and business intelligence tools, organizations can now leverage the data they collect to run more efficiently, enable better decision-making, and become more agile and scalable. Today, optimizing access and throughput to data is critical because there is more data volume to track. It is critical to have a platform that can deliver the performance, scale, and agility that businesses need to grow over time.

The self-driving database is poised to provide a significant boost to these capabilities. Because self-driving databases automate expensive, time-consuming manual processes, they free up business users to become more proactive with their data. By having direct control over creating and using databases, users gain control and autonomy while still maintaining essential security standards.

3.4.8 Database challenges

Today's large enterprise databases often support very complex queries and are expected to deliver nearly instant responses to those queries. As a result, database administrators are constantly called upon to employ various methods to help improve performance. Addressing all of these challenges

can be time-consuming and can prevent database administrators from performing more strategic functions. Some common challenges that they face include:

3.4.8.1 Absorbing significant increases in data volume:

The explosion of data coming in from sensors, connected machines, and dozens of other sources keeps database administrators scrambling to efficiently manage and organize their companies' data.

3.4.8.2 Ensuring data security:

Data breaches are happening everywhere these days, and hackers are getting more inventive. It is more important than ever to ensure that data is secure but also easily accessible to users.

3.4.8.3 Keeping up with demand:

In today's fast-moving business environment, companies need real-time access to their data to support timely decision-making and to take advantage of new opportunities.

3.4.8.4 Managing and maintaining the database and infrastructure:

Database administrators must continually watch the database for problems and perform preventative maintenance and the application of software upgrades and patches. As databases become more complex and data volumes grow, companies are faced with the expense of hiring additional talent to monitor and tune their databases.

3.4.8.5 Removing limits on scalability:

A business needs to grow if it is going to survive, and its data management must grow along with it. However, it is challenging for database administrators to predict how much capacity the company will need, particularly with on-premises databases.

3.4.8.6 Ensuring data residency, data sovereignty, or latency requirements:

Some organizations have use cases that are better suited to run on-premises. In those cases, engineered systems that are pre-configured and pre-optimized for running the database are ideal. Customers achieve higher availability, more excellent performance and up to 40% lower cost with Oracle Exadata, according to Wikibon's recent analysis (PDF).

3.5 Examples of web-based CMMS

3.5.1 Smartsheet Software [25]

Smartsheet is a work execution platform and collaboration tool with a familiar spreadsheet-like interface that helps teams plan, track, and manage projects in real-time. Smartsheet features include various project management tools, such as document and resource management, project reporting, task management, file sharing, and timeline tracking. Smartsheet can help teams standardize a project process, maximize productivity, and improve collaboration with automated workflow options that fit individual work preferences. Data can be sorted using a card, grid, Gantt chart, and calendar view within the platform's project management tools. With a real-time dashboard, teams can track project status, manage resources, and assign tasks to improve overall visibility and work efficiency. To keep everyone on the same page, project plan information can be shared with leadership team members and key stakeholders. Activity log reports and customizable dashboards help ensure that team collaboration is consistent across projects of any size and kind, from software development to marketing plans. With the Smartsheet app, teams can access project information and workflow data while on the go. The mobile app is compatible with iOS and Android devices.

Figure 3.5: Smartsheet Software

3.5.2 UpKeep [28]

UpKeep's CMMS is a modern maintenance and asset management solution for a team. From a desktop to a phone and even a tablet, UpKeep is easily accessible from anywhere at any time. Create work orders on the go, get notifications when tasks are updated, and receive alerts straight from the app when assets go down, making the business run more efficiently than ever before. UpKeep offers core maintenance functionality, such as asset, inventory, work order management, and preventive maintenance, to create service schedules. Users can see a tasks overview page showing upcoming work and due dates, criticality and assets or workers assigned to each. Users can include notes, a colour-coded priority rating, images, assets, and users when adding new tasks. The mobile application gives users the ability to create projects, assign work orders, manage assets and more. UpKeep is designed for small to midsize companies across various industry verticals. UpKeep offers services on a monthly subscription basis that includes support via phone, email and through an online knowledge base.

Figure 3.6: UpKeep Software

3.5.3 Asset Panda Software [3]

Asset Panda is a cloud-based platform for facility managers that offers applications, including asset tracking and maintenance management. It is compatible with Windows, Mac, iPad, and iPhone and can be used anywhere while updating real-time data. Asset Panda helps users assign contacts by location and track depreciation. The user can use a barcode scanner to look up asset details and automate pick lists. Role-based security ensures that employees are granted the appropriate level of access, so they always see the information relevant to what they are working on. Asset Panda offers customized exporting and reporting features. Reports can be automated via email, adding custom calculation fields (and designate if each field should be required) before sending. Asset Panda helps users in auditing, facilities management, equipment support ticketing, compliance and purchase order management. Other features include asset photo tagging, replication configuration and Gantt charts. Support is available via chat, email and phone. Pricing is either per asset or per user.

Figure 3.7: Asset Panda Software

3.5.4 EZOfficeInventory Software [10]

EZOfficeInventory is a dynamic asset and maintenance management solution tailored for companies of all sizes. The cloud-based software works as the user do and is accessible from anywhere and at any time. Track items across locations and departments, scan barcodes right from the user's phone, streamline item maintenance to boost functionality, maintain optimized stock levels at all times, and manage all the users and vendors with one solution. The inventory management module enables users to track inventory levels with QR codes and barcode tags. EZOfficeInventory features a mobile app with barcode and QR code scanning functionality, so the user can track his inventory wherever he may be. GPS tracking enables users to track item locations to monitor their assets on an interactive map and identify lost or misplaced assets. The solution offers a REST-based API that integrates with other REST-based systems. Customized reporting and dashboards help users track and monitor assets. Users can analyze historical inventory data and make decisions regarding their inventory management practices. The solution also integrates with third-party applications such as Zendesk, Jira, Dropbox, Active Directory, and OneLogin. EZOfficeInventory offers both monthly and annual subscriptions that include support via phone, email and through an online knowledge base.

Figure 3.8: EZOfficeInventory Software

3.5.5 Fiix Software [11]

Fiix is a cloud-based computerized maintenance management system (CMMS) that helps businesses organize their maintenance departments, get on top of maintenance backlog, and work towards preventive maintenance. The CMMS helps manage work orders and physical assets, schedules and tracks maintenance, and keeps detailed records of asset performance and maintenance history. Other features include inventory tracking, parts and supplies management, an interactive calendar, printable QR codes for easy asset tagging, customizable reports, multi-site management, ERP integrations and more. Fiix is a web-based solution, aiding in setup and automatic updates. All users also get access to the Fiix mobile app, which puts the CMMS into the hands of technicians in the field. The system is compatible with Mac and Windows operating systems and any iOS or Android mobile device.

Figure 3.9: Fiix Software

3.6 Conclusion

In this chapter we spoke about the client-server architecture, its types and some examples, and then we took a look on websites, its history, its creation, its structure and its types, and then discovered the database and its types, and we saw some examples of web-based CMMS. In the next chapter we will talk about our design and production and resume our work in some diagrams.

Chapter 4

Design and Production

4.1 Introduction

In this chapter we will disccuss what client-server architecture we will have in our website, as well as the definition of three diagramms (usecase, sequence and class diagramm), and we will use those diagramms to demonstrate all of our website.

4.2 Our Architecture

The architecture that we used in our website is the 3-tier architecture, the server waits for requests from the frontend and processes them and then sends a response to the frontend and then shows it to the user, all of this happens in less than a second to ensure the best user experience

4.3 Unified modeling language (UML diagrams) [16]

4.3.1 usecase diagram [29] [13]

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams. Either circles or ellipses represent the use cases. The actors are often shown as stick figures.

4.3.2 class diagrams [8] [20]

A class diagram is a diagram used in designing and modeling software to describe classes and their relationships. Class diagrams enable us to model software in a high level of abstraction without looking at the source code. Classes in a class diagram correspond with classes in the source code. The diagram shows the names and attributes of the classes, connections between the classes, and sometimes also the methods of the classes.

4.3.3 sequence diagram [24]

A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction

4.4 LucidChart [17]

Lucidchart is a web-based proprietary platform that allows users to collaborate on drawing, revising, and sharing charts and diagrams. It is produced by Lucid Software Inc., based in Utah, United States.

4.5 Our Class Diagram

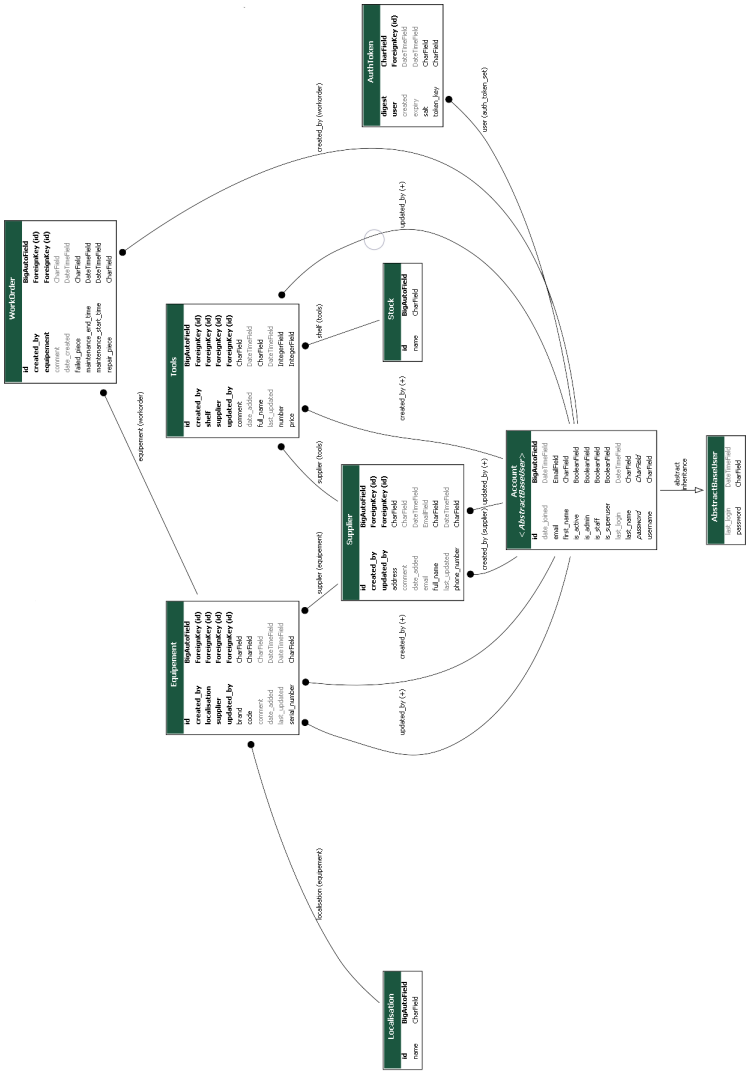

Figure 4.1: Our Class Diagram

4.6 Our Sequence Diagram

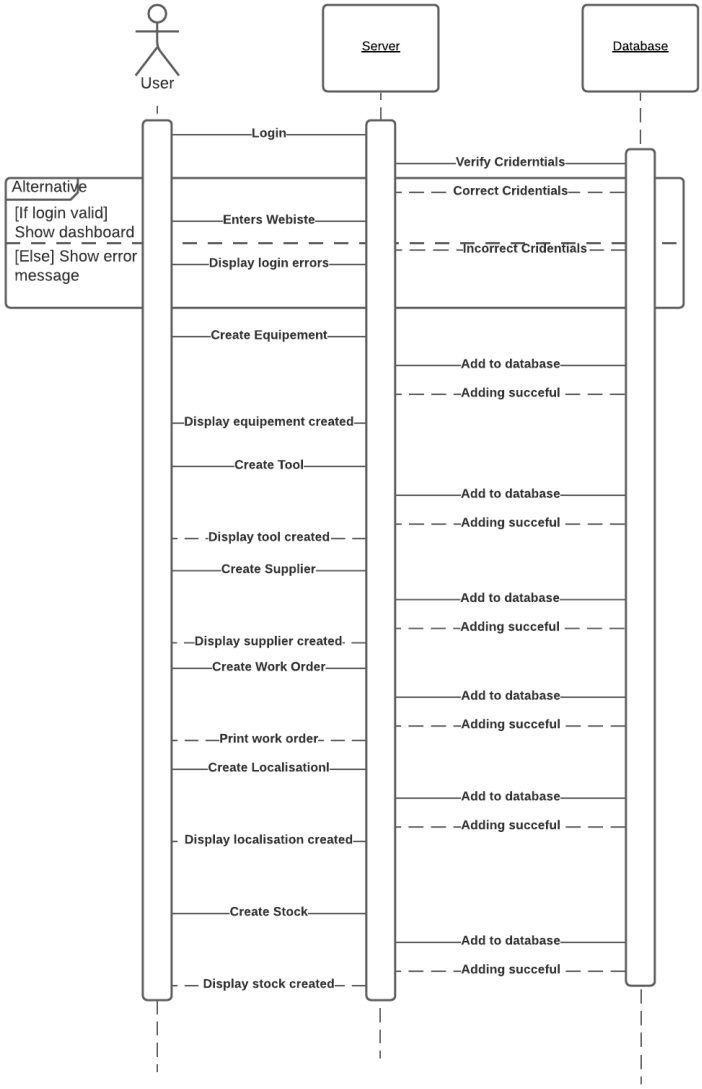

Figure 4.2: Our Sequence Diagram

4.7 Our UseCase Diagram

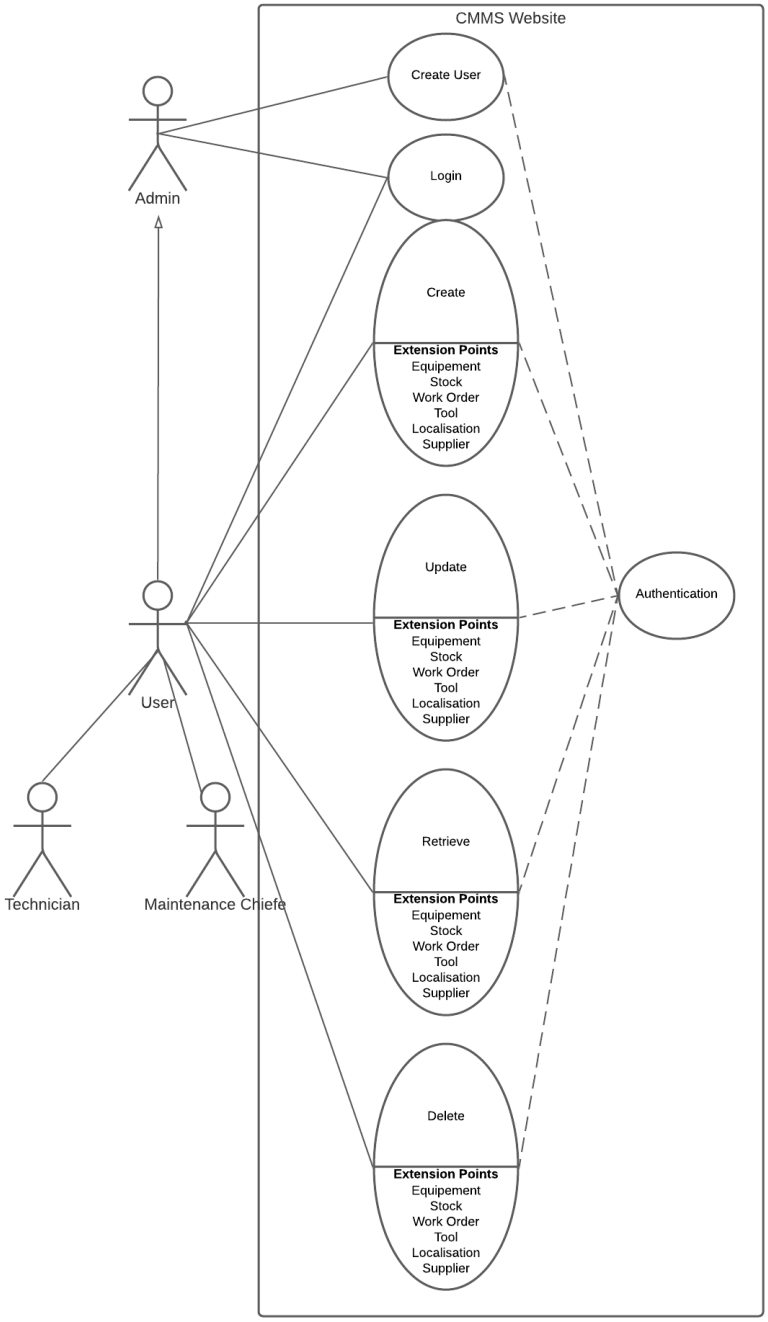

Figure 4.3: Our UseCase Diagram

4.8 Conclusion

We have discussed in this chapter the UML diagrams, and created our own diagramms to further demonstrate our website, and in the next chapter we will talk about every programming language

that we used in our website, and try to build our website and see all of its functionalities.

Chapter 5

Implementation And Simulation

5.1 Introduction

In this chapter we will discover in a shallow way all the programming languages that we used in our website, and then we will discover all the functionalities of our website, the roles of all the users, inputs and outputs and all the rest of the pages.

5.2 Programming Languages Used

5.2.1 Python [35] [30]

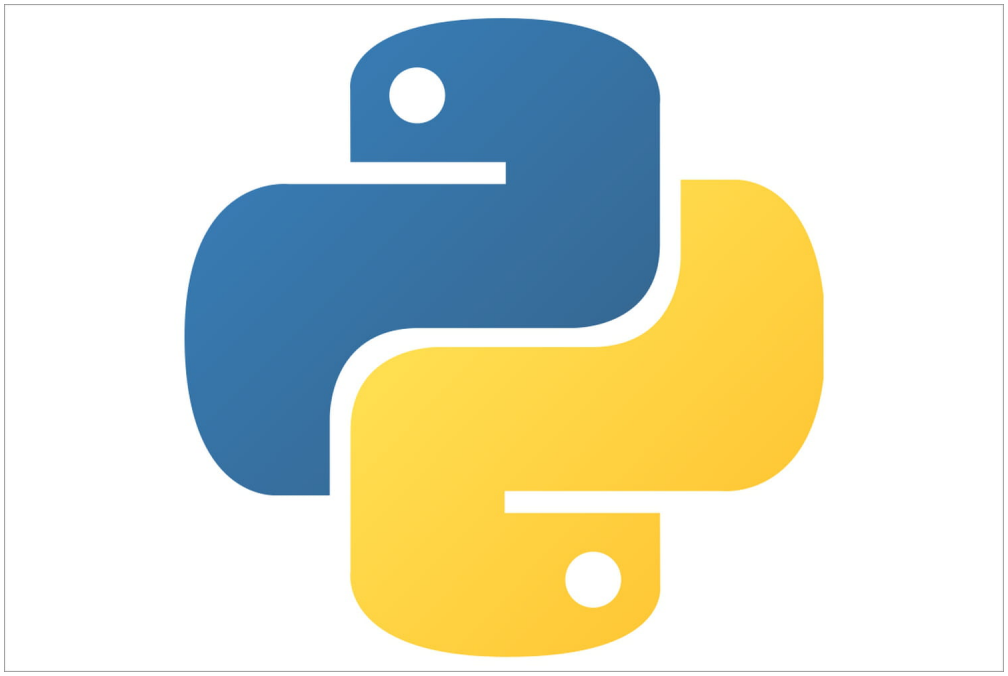

Figure 5.1: Python's Logo

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development and use as a scripting or glue language to connect existing components. Python's simple, easy-to-learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is straightforward: a bug or wrong input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program does not catch the exception, the interpreter prints a stack trace. A source-level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

5.2.2 Django [12]

Figure 5.2: Django's Logo

5.2.2.1 What is Django?

Django is a free and open-source web application framework, written in Python. A web framework is a set of components that helps the user to develop websites faster and easier.

When the user is building a website, he always needs a similar set of components: a way to handle user authentication (signing up, signing in, signing out), a management panel for your website, forms, uploading files, and many others.

Luckily, some people long ago noticed that web developers face similar problems when building a new site, so they teamed up and created frameworks (Django being one of them) that give ready-made components to use.

Frameworks exist to save the user from reinventing the wheel and help alleviate some of the overhead when building a new site.

5.2.2.2 Why do we need a framework? [34]

To understand what Django is for, we need to take a closer look at the servers. The first thing is that the server needs to know that the user wants it to serve him a web page.

Imagine a mailbox (port) that is monitored for incoming letters (requests). A web server does it. The web server reads the letter and then sends a response with a webpage. However, when the user wants to send something, he needs to have some content. Moreover, Django is something that helps the user creates the content.

5.2.2.3 What happens when someone requests a website from a server?

When a request comes to a web server, it is passed to Django, which tries to figure out what is requested. It takes a web page address first and tries to figure out what to do. Django's urlresolver does this part (note that a website address is called a URL – Uniform Resource Locator – so the name urlresolver makes sense). It is not very smart – it takes a list of patterns and tries to match the URL. Django checks patterns from top to bottom, and if something is matched, then Django passes the request to the associated function (it is called view).

Imagine a mail carrier with a letter. He is walking down the street and checks each house number against the one on the letter. If it matches, he puts the letter there. This is how the urlresolver works!

All the exciting things are done in the view function: we can look at a database to look for some information. Maybe the user asked to change something in the data, Like a letter saying, "Please change the description of my job." The view can check if the user is allowed to do that, then update the job description for him and send back a message: "Done!" Then the view generates a response, and Django can send it to the user's web browser.

5.2.3 JavaScript [22]

Figure 5.3: JavaScript's Logo

5.2.3.1 What is JavaScript?

JavaScript is a text-based programming language used both on the client-side and server-side that allows the user to make web pages interactive. Where HTML and CSS give structure and style to web pages, JavaScript gives web pages interactive elements that engage a user.

Common examples of JavaScript that the user might use every day include the search box on Amazon, a news recap video embedded on The New York Times, or refreshing his Facebook feed.

Incorporating JavaScript improves the user experience of the web page by converting it from a static page into an interactive one. To recap, JavaScript adds behavior to web pages.

5.2.3.2 What is JavaScript used for? [14]

JavaScript is mainly used for web-based applications and web browsers. However, JavaScript is also used beyond the Web in software, servers, and embedded hardware controls. Here are some basic things JavaScript is used for:

1. **Adding interactive behavior to web pages:**

   JavaScript allows users to interact with web pages. There are almost no limits to the things the user can do with JavaScript on a web page, and these are just a few examples:

   - Show or hide more information with the click of a button
   - Change the color of a button when the mouse hovers over it
   - Slide through a carousel of images on the homepage
   - Zooming in or zooming out on an image
   - Displaying a timer or count-down on a website
   - Playing audio and video on a web page
   - Displaying animations
   - Using a drop-down hamburger menu

2. **Creating web and mobile apps**

   Developers can use various JavaScript frameworks for developing and building Web and mobile apps. JavaScript frameworks are collections of JavaScript code libraries that provide developers with pre-written code for standard programming features and tasks—literally a framework to build websites or web applications around.

   Popular JavaScript front-end frameworks include React, React Native, Angular, and Vue. Many companies use Node.js, a JavaScript runtime environment built on Google Chrome's JavaScript V8 engine. A few famous examples include PayPal, Linked In, Netflix, and Uber.

3. **Building web servers and developing server applications**

   Beyond websites and apps, developers can also use JavaScript to build simple web servers and develop the back-end infrastructure using Node.js.

4. **Game development**

   Of course, JavaScript can also be used to create browser games. These are excellent ways for beginning developers to practice their JavaScript skills.

5.2.3.3 Why use JavaScript over other programming languages?

Aside from the unlimited possibilities, there are many reasons for web developers to use JavaScript over other programming languages:

- JavaScript is the only programming language native to the web browser.
- JavaScript is the most popular language.
- There is a low threshold to get started.
- It is a fun language to learn.

5.2.4 HTML [7]

Figure 5.4: HTML's Logo

5.2.4.1 What is HTML?

Hypertext Markup Language (HTML) is a computer language that makes up most web pages and online applications. A hypertext is a text used to reference other pieces of text, while a markup language is a series of markings that tells web servers the style and structure of a document.

HTML is not considered a programming language as it cannot create dynamic functionality. Instead, with HTML, web users can create and structure sections, paragraphs, and links using elements, tags, and attributes.

Here are some of the most common uses for HTML:

- **Web Development:** Developers use HTML code to design how browsers display web page elements, such as text, hyperlinks, and media files.
- **Internet Navigation:** Users can easily navigate and insert links between related pages and websites as HTML is heavily used to embed hyperlinks.
- **Web Documentation:** HTML makes it possible to organize and format documents, similarly to Microsoft Word.

It is also worth noting that HTML is now considered an official web standard. The World Wide Web Consortium (W3C) maintains and develops HTML specifications and provides regular updates.

This article will go over the basics of HTML, including how it works, its pros and cons, and how it relates to CSS and JavaScript.

5.2.4.2 How Does HTML Work?

The average website includes several different HTML pages. For instance, a home page, an about page, and a contact page would all have separate HTML files.

HTML documents are files that end with a .html or .htm extension. A web browser reads the HTML file and renders its content so that internet users can view it.

All HTML pages have a series of HTML elements, consisting of a set of tags and attributes. HTML elements are the building blocks of a web page. A tag tells the web browser where an element begins and ends, whereas an attribute describes the characteristics of an element.

The three main parts of an element are:

- **Opening tag:** used to state where an element starts to take effect; the tag is wrapped with opening and closing angle brackets. For example, use the start tag to create a paragraph.
- **Content:** this is the output that other users see.
- **The closing tag:** is the same as the opening tag but with a forward slash before the element name. For Example, to end a paragraph.

The combination of these three parts will create an HTML element:

```
<p>This is how you add a paragraph in HTML.</p>
```

Another critical part of an HTML element is its attribute, which has two sections, a name and attribute value. The name identifies the additional information a user wants to add, while the attribute value gives further specifications.

For example, a style element adding the color purple and the font-family Verdana will look like this:

```
<p style="color:purple;font-family:verdana">This is how you add a paragraph in HTML.</p>
```

Another attribute, the HTML class, is most important for development and programming. The class attribute adds style information that can work on different elements with the same class value.

For example, we will use the same style for a heading <h1> and a paragraph . The style includes background color, text color, border, margin, and padding, under the class .important. To achieve the same style between <h1> and , add class='important' after each start tag:

```
<html>
<head>
<style>
.important {
background-color: blue;
color: white;
border: 2px solid black;
margin: 2px;
padding: 2px;
}
</style>
</head>
<body>
<h1 class="important">This is a heading</h1>
<p class="important">This is a paragraph.</p>
</body>
</html>
```

Most elements have an opening and a closing tag, but some do not need closing tags to work, such as empty elements. These elements do not use an end tag because they do not have content:

```
<img src="/" alt="Image">
```

This image tag has two attributes – an src attribute, the image path, and an alt attribute, the descriptive text. However, it does not have content nor an end tag.

Lastly, every HTML document must start with a <!DOCTYPE> declaration to inform the web browser about the document type. With HTML5, the doctype HTML public declaration will be:

```
<!DOCTYPE html>
```

5.2.5 CSS [33]

Figure 5.5: CSS's Logo

5.2.5.1 What is CSS?

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, the user can control the color of the text, style of fonts, spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, control the display in different devices, and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand, but it provides robust control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

5.2.5.2 Advantages of CSS

- **CSS saves time:** CSS can be written once and then reuse the same sheet in multiple HTML pages. A user can define a style for each HTML element and apply it to as many Web pages as the user wants.
- **Pages load faster:** If a user uses CSS, he does not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So, less code means faster download times.
- **Easy maintenance:** To make a global change, change the style, and all elements in all the web pages will be updated automatically.
- **Superior styles to HTML:** CSS has a much wider array of attributes than HTML so that a user can give a far better look to his HTML page in comparison to HTML attributes.
- **Multiple Device Compatibility:** Style sheets allow content to be optimized for more than one type of device. Different versions of a website can be presented for handheld devices such as PDAs and cell phones or printing by using the same HTML document.
- **Global web standards:** Now HTML attributes are being deprecated, and CSS is recommended. So it is good to start using CSS in all the HTML pages to make them compatible with future browsers.

5.2.5.3 Who Creates and Maintains CSS?

CSS is created and maintained through a group of people within the W3C called the CSS Working Group. The CSS Working Group creates documents called specifications. When a specification has been discussed and officially ratified by the W3C members, it becomes a recommendation.

These ratified specifications are called recommendations because the W3C has no control over the actual implementation of the language. Independent companies and organizations create that software.

NOTE: The World Wide Web Consortium or W3C is a group that makes recommendations about how the Internet works and how it should evolve.

5.2.6 ReactJS [1]

Figure 5.6: ReactJS's Logo

5.2.6.1 What is ReactJS?

React is the most popular front-end JavaScript library in the field of web development. It is used by large, established companies and newly-minted startups alike (Netflix, Airbnb, Instagram, and the New York Times, to name a few). React brings many advantages to the table, making it a better choice than other frameworks like Angular.js.

React is a JavaScript library created for building fast and interactive user interfaces for web and mobile applications. It is an open-source, component-based, front-end library responsible only for the application's view layer. In Model View Controller (MVC) architecture, the view layer is responsible for the app's looks and feels. React was created by Jordan Walke, a software engineer at Facebook.

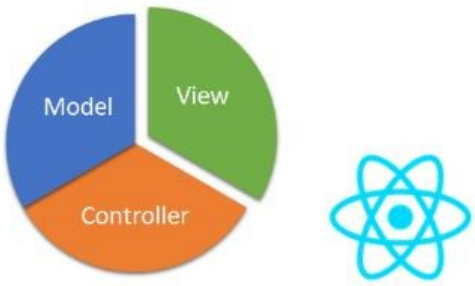

Figure 5.7: ReactJS MVC architecture

An Instagram webpage, for example, is entirely built using React to get a better understanding of how React works. As the illustration shows, React divides the UI into multiple components, making the code easier to debug. This way, each component has its property and function.

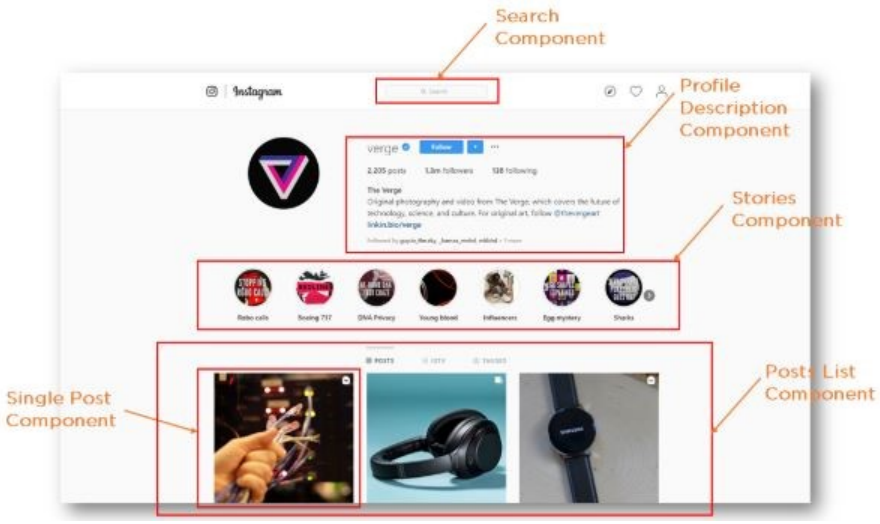

Figure 5.8: Instagram Components

5.2.6.2 Why React?

React's popularity today has eclipsed that of all other front-end development frameworks. Here is why:

- **Easy creation of dynamic applications:** React makes it easier to create dynamic web applications because it requires less coding and offers more functionality than JavaScript, where coding often gets complex very quickly.
- **Improved performance:** React uses Virtual DOM, thereby creating web applications faster. Virtual DOM compares the components' previous states and updates only the items in the Real DOM that were changed instead of updating all of the components again, as conventional web applications do.
- **Reusable components:** Components are the building blocks of any React application, and a single app usually consists of multiple components. These components have their logic and controls, and they can be reused throughout the application, which dramatically reduces the application's development time.
- **Unidirectional data flow:** React follows a unidirectional data flow. This means that when designing a React app, developers often nest child components within parent components. Since the data flows in a single direction, it becomes easier to debug errors and know where a problem occurs in an application at the moment in question.
- **Small learning curve:** React is easy to learn, as it mainly combines basic HTML and JavaScript concepts with some beneficial additions. Still, as is the case with other tools and frameworks, the user has to spend some time properly understanding React's library.
- **It can be used to develop both web and mobile apps:** We already know that React is used to develop web applications, but that is not all it can do. A framework called React Native, derived from React itself, is hugely popular and used for creating beautiful mobile applications. So, in reality, React can be used for making both web and mobile applications.
- **Dedicated tools for easy debugging:** Facebook has released a Chrome extension that can be used to debug React applications. This makes the process of debugging React web applications faster and easier. The above reasons more than justify the popularity of the React library and why it is being adopted by a large number of organizations and businesses.

5.2.6.3 Features of React [26]

Figure 5.9: React Features

1. **JSX - JavaScript Syntax Extension:** JSX is a syntax extension to JavaScript. It is used with React to describe what the user interface should look like. By using JSX, we can write HTML structures in the same file that contains JavaScript code. This makes the code easier to understand and debug, as it avoids using complex JavaScript DOM structures.

```
const name = 'Simplilearn';
const greet = <h1>Hello, {name}</h1>;
```

   The above code shows how JSX is implemented in React. It is neither a string nor HTML. Instead, it embeds HTML into JavaScript code.

2. **Virtual DOM:** React keeps a lightweight representation of the "real" DOM in the memory, and that is known as the "virtual" DOM (VDOM). Manipulating real DOM is much slower than manipulating VDOM because nothing gets drawn on the screen. When the state of an object changes, VDOM changes only that object in the real DOM instead of updating all the objects.

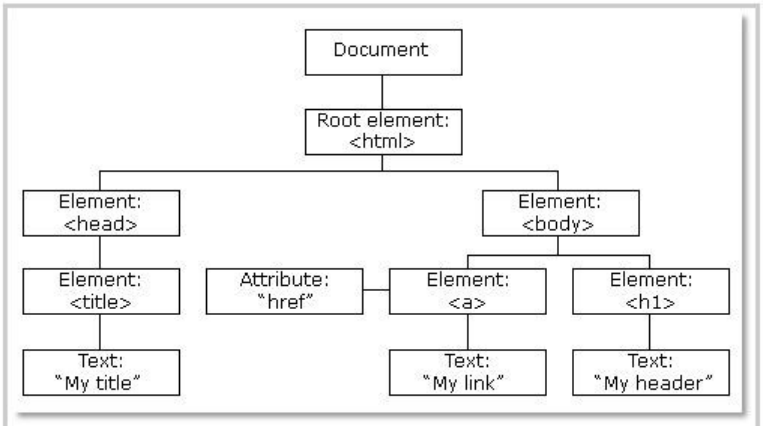

Figure 5.10: DOM of a Webpage

DOM (Document Object Model) treats an XML or HTML document as a tree structure in which each node is an object representing a part of the document.

Figure 5.11: Virtual DOM

When the state of an object changes in a React application, VDOM gets updated. It then compares its previous state and then updates only those objects in the real DOM instead of updating all of the objects. This makes things move fast, especially when compared to other front-end technologies that update each object even if only a single object changes in the web application.

3. **Performance:** React uses VDOM, making web applications run much faster than those developed with alternate front-end frameworks. React breaks a complex user interface into individual components, allowing multiple users to work on each component simultaneously, speeding up the development time.

4. **Extensions:** React goes beyond simple UI design and has many extensions that offer complete application architecture support. It provides server-side rendering, which entails rendering a normally client-side only web application on the server, and then sends a fully rendered page to the client. It also employs Flux and Redux extensively in web application development. Finally, React Native, a popular framework derived from React, is used to create cross-compatible mobile applications.

5. **One-way Data Binding:** React's one-way data binding keeps everything modular and fast. A unidirectional data flow means that when developers design a React app, they often nest child components within parent components. This way, a developer knows where and when an error occurs, giving them better control of the whole web application.

Figure 5.12: One-way data binding

6. **Debugging:** React applications are easy to test due to a large developer community. Facebook even provides a tiny browser extension that makes React debugging faster and easier.

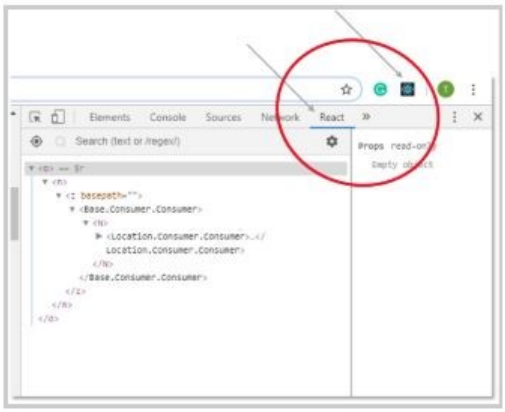

Figure 5.13: React Extension

This extension, for example, adds a React tab in the developer tools option within the Chrome web browser. The tab makes it easy to inspect React components directly.

5.3 The Website

5.3.1 Authentication

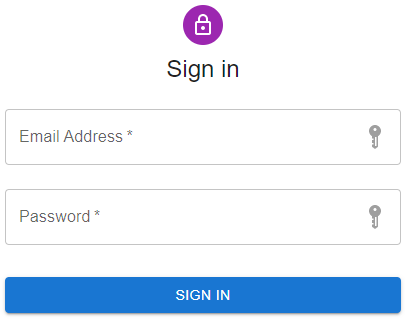

Figure 5.14: Login Page

In 5.14 we find the sign in page so that all the website users can authenticate and get access to it.

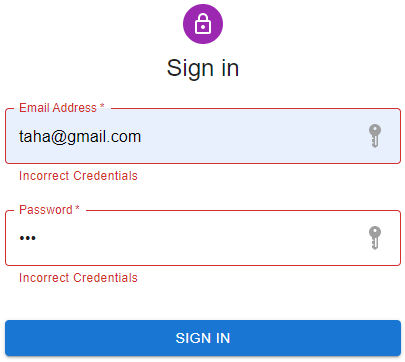

Figure 5.15: Login Page With Errors

In 5.15 authentication errors appeared when the user enter wrong login cridentials.

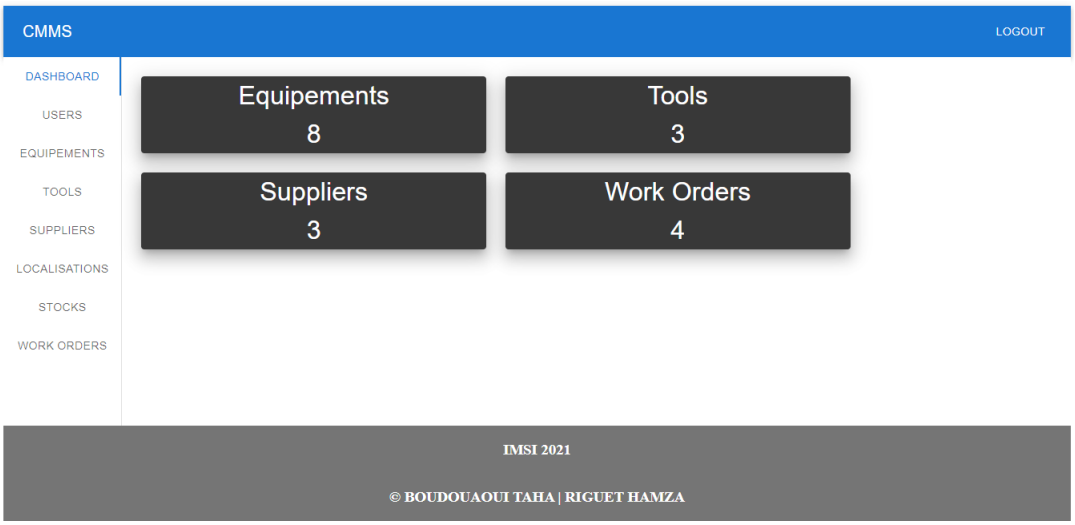

Figure 5.16: Dashboard

When the user authenticate succefully he gets access to the website and the first page is the dashboard 5.16, where we can find how many Equipements, Tools, Suppliers and Work Orders we have There is also a simple navigation bar with a logout button in the top, and a footer in the bottom of the page with the credits. and on the left we can see the Tabs to access every thing a user needs.

5.3.2 User Panel

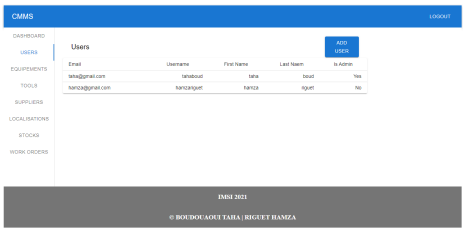

Figure 5.17: Admin User.

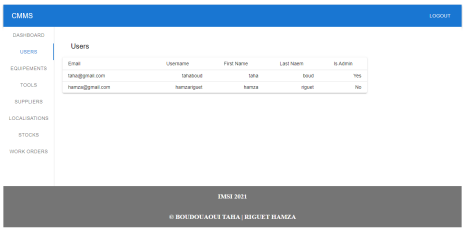

Figure 5.18: Normal User.

On the users panel we can see all the users listed, and in this page the admin 5.17 can find the Add User button to add a user.

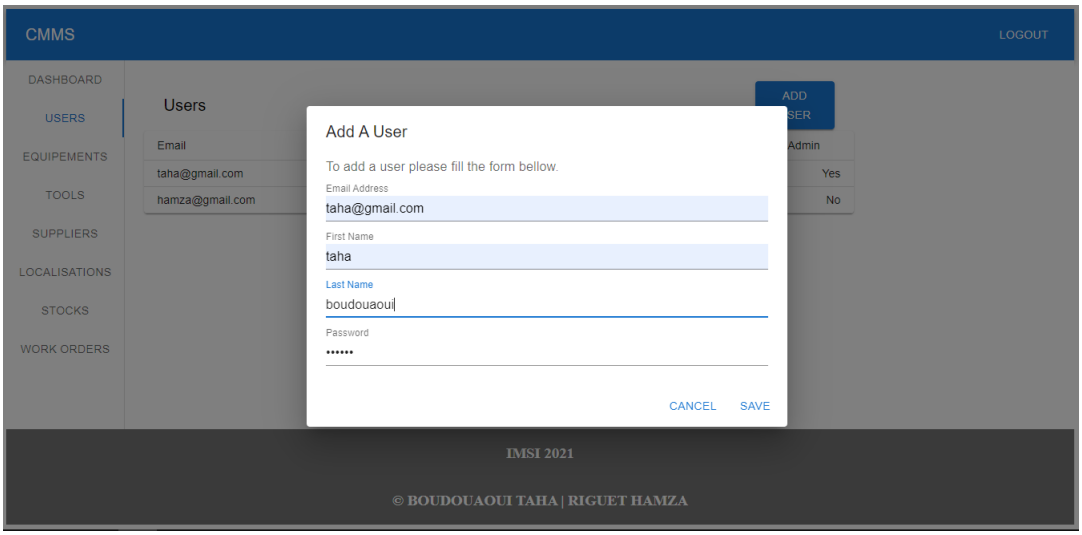

Figure 5.19: Add User Dialog

The admin can input the new user's email, first name, last name and password, and then save the new user 5.19.

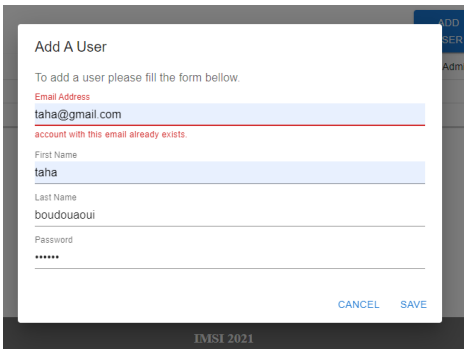

Figure 5.20: Add User Dialog With Error

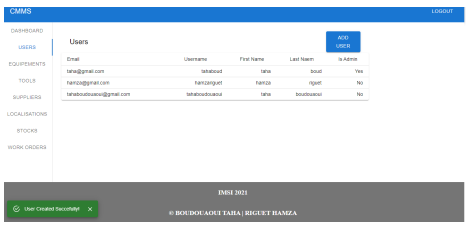

Figure 5.21: User Added Succefully

After saving the website either respond with an error like in 5.20, the error states that a user

with that email already exists. Or respond with a small success alert in the bottom left in the page saying "User Created Succefully" like in 5.21, and the new user is added in the bottom of the list.

5.3.3 Localisation Panel

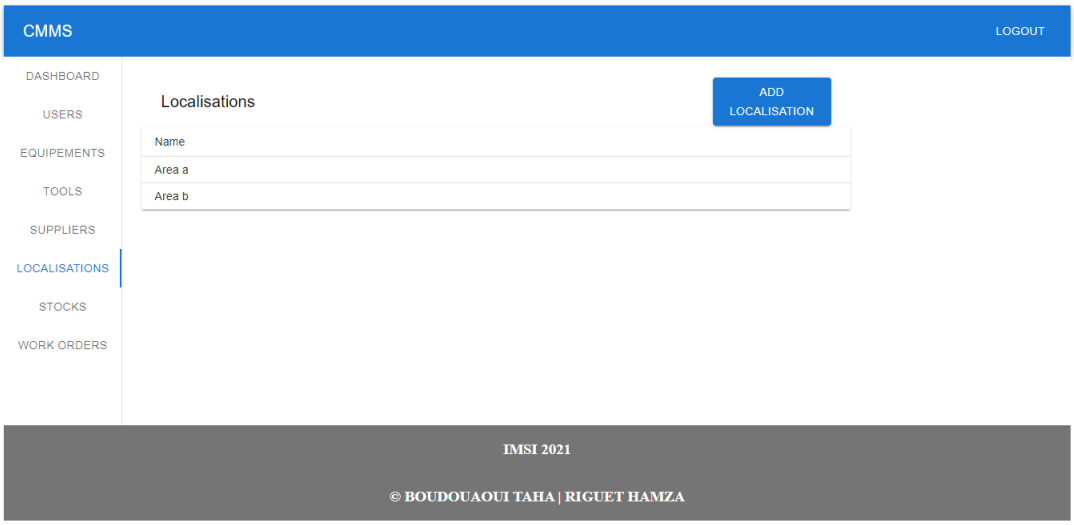

Figure 5.22: Localisation Panel

In the localisation panel 5.22 we can see all the localisations listed which they only have a name field and any user can add a localisation by clicking on the 'Add Localisation' button.

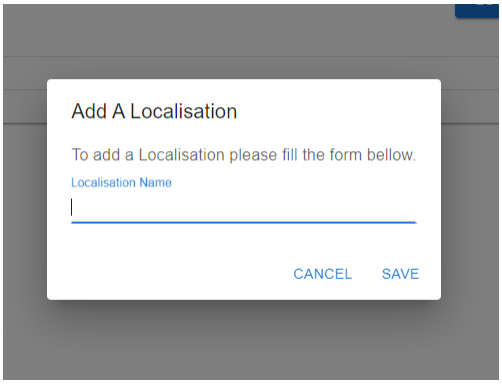

Figure 5.23: Add Localisation Dialog

By clicking on the 'Add Localisation', an 'Add Localisation Dialog' will pop up to insert the new localisation's name and then save it.

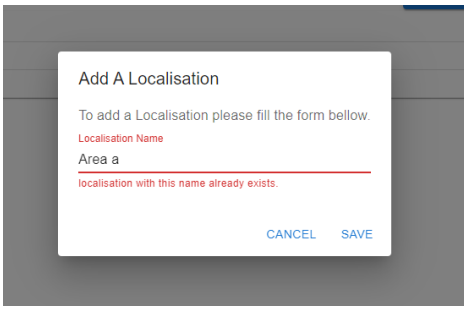

Figure 5.24: Add Localisation Dialog With Error

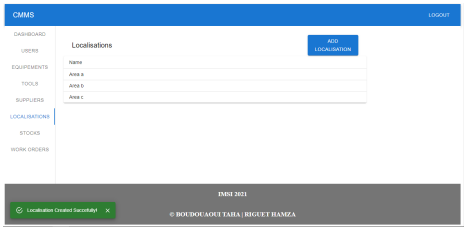

Figure 5.25: Localisation Added Succefully

After clicking save, a response from the server will either tell you that there is already a localisation with the same name 5.25, or a success message 'Localisatioin Added Succefully' will show in the bottom left of the screen and the new localisation is in the bottom of the list 5.25.

5.3.4 Supplier Panel

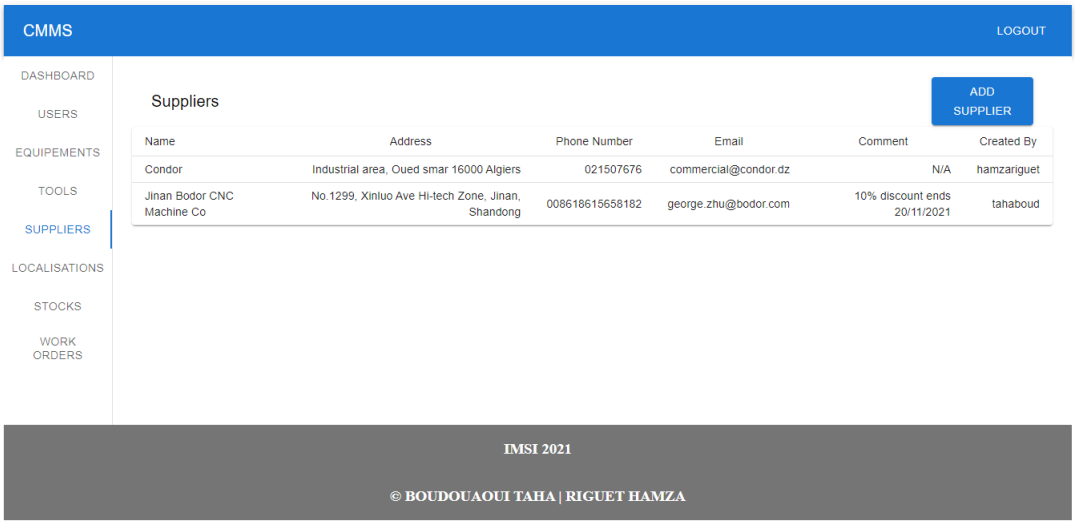

Figure 5.26: Supplier Panel

In the supplier's panel 5.26 we can see all the suppliers we have listed with their name, address, phone number, email , a comment for any additional informations and the user that added the supplier's informations, we can also see some fields have "N/A" in them which indicates that there was no informations added as the email and comment fields are optional.

And to add a new supplier we can click on the 'Add Supplier' button.

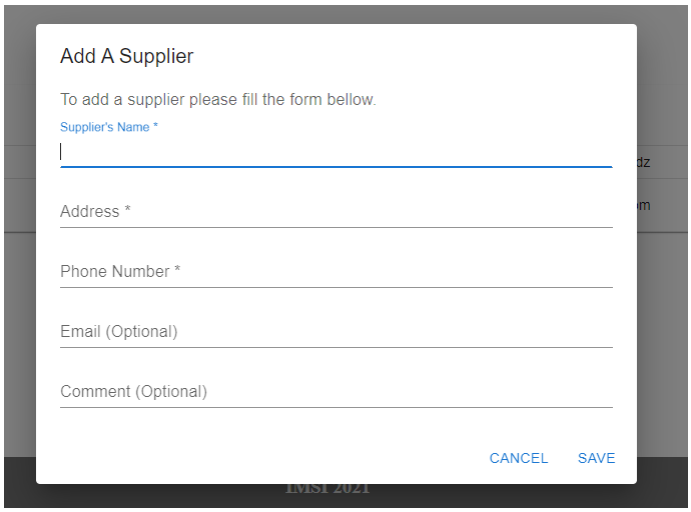

Figure 5.27: Add Supplier Dialog

After clicking on the 'Add Supplier' button a dialog will pop up with a form to add the new supplier 5.27, it contains a name, address, phone number, email and comment fields, the obligatory fields are marked with a star next to their label, and the other ones (email, comment) are optional.

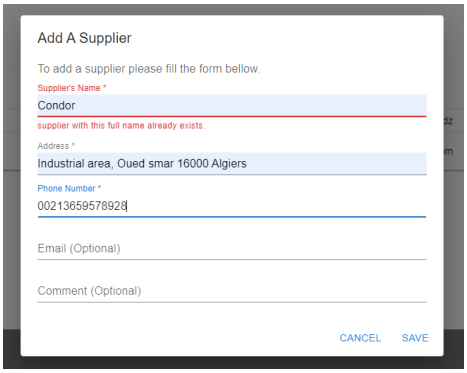

Figure 5.28: Add Supplier Dialog With Error

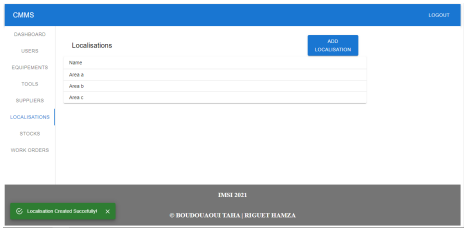

Figure 5.29: Supplier Added Succefully

In the first picture 5.28 the user tried to add an already existing supplierso the server responded with the error 'Supplier with this name already exists'. And in the second picture 5.29 the user added a new supplier 'siemens' which displayed a success message in the bottom left 'Supplier Added Succefully' and its information is displayed in the bottom of the table.

5.3.5 Equipement Panel

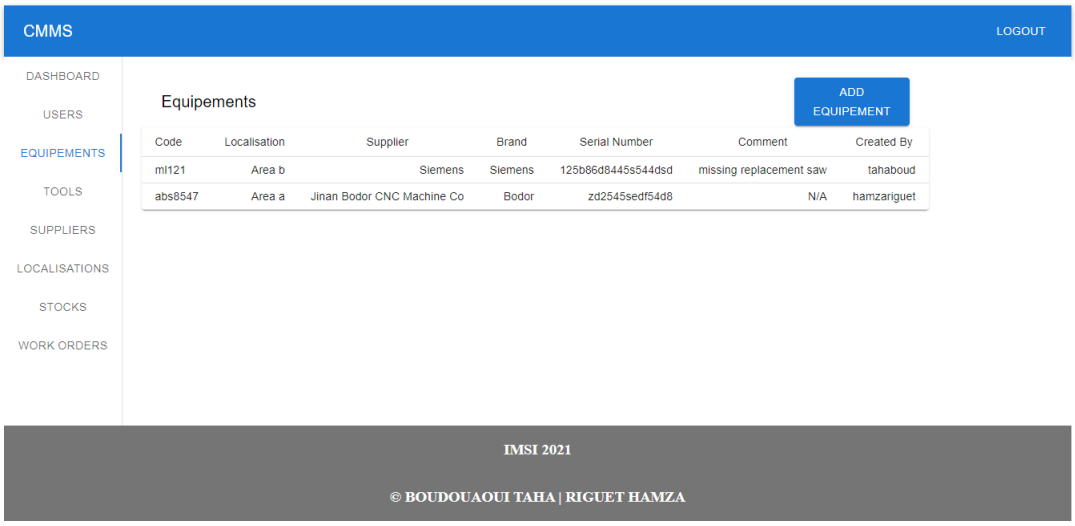

Figure 5.30: Equipement Panel

In the equipement panel 5.30 we can see all the equipements listed in the table, it contains a code, localisation, supplier, brand, serial number, comment and the user that added the equipement.
And to add a new equipement we click on the 'Add Equipement' button.

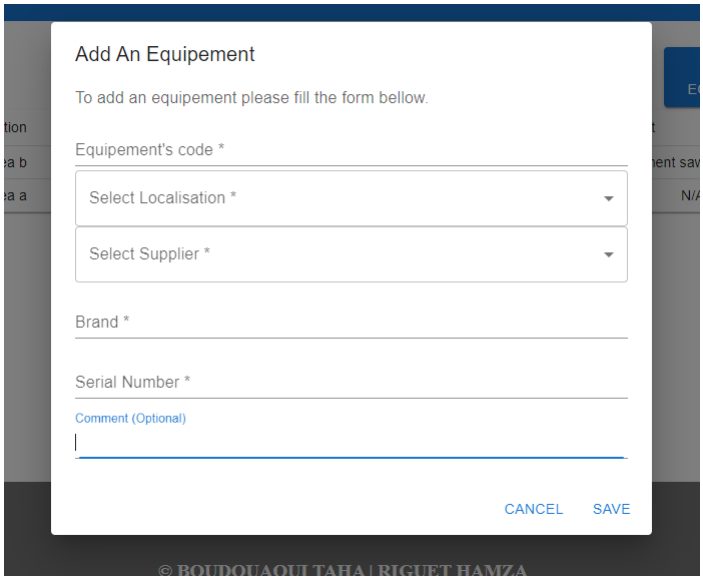

Figure 5.31: Add Equipement Dialog

After clicking on the 'Add Equipement' button a dialog will pop up with a form to add the new equipement 5.31, it contains a field for code, a select field for both the localisation the new equipement is in and the supplier the equipement was bought from, a brand field, a serial number field and a comment for any additional informations.

After inputing all the information needed the user can click save to finish adding the new equipement to the database, which will display one of two types of server replies.

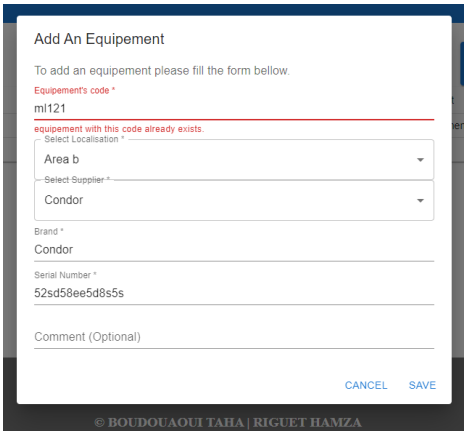

Figure 5.32: Add Equipement Dialog With Error

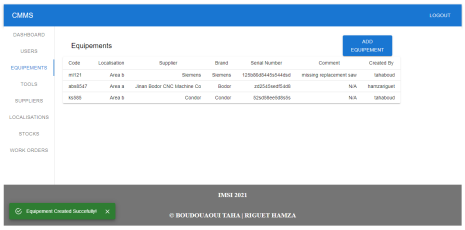

Figure 5.33: Equipement Added Succefully

In the first picture 5.32 we can see the server responded with an error telling the user that an equipement with that code already exists.

And in the second image 5.33 the server added the new equipement and responded with a success message in the bottm left of the page, and we can see the new equipement informations displayed in the bottom of the table.

5.3.6 Stock Panel

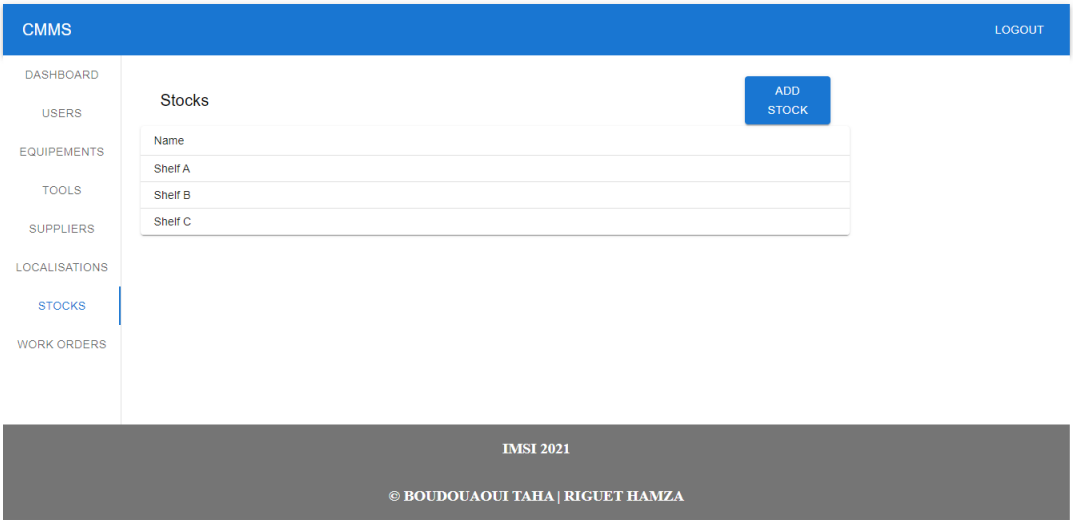

Figure 5.34: Stock Panel

In the stocks panel 5.34 we can see all the places in stock listed in the table, described by their name.

And to add a new stock the user must press the 'Add Stock' button

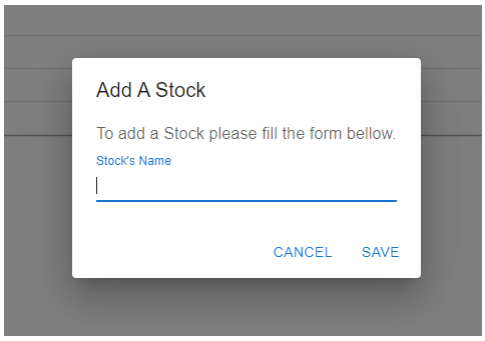

Figure 5.35: Stock Dialog

After clicking on the 'Add Stock' button, a dialog will pop up with a form containing a name field to add a new stock.

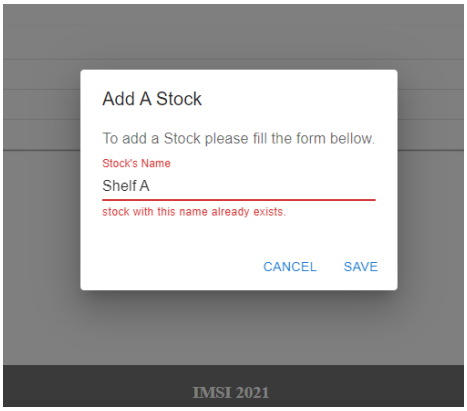

Figure 5.36: Add Stock Dialog With Error

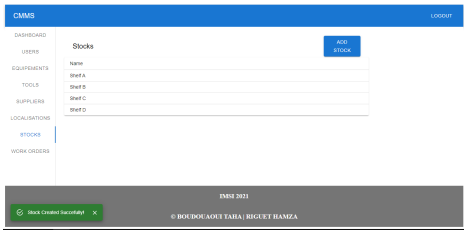

Figure 5.37: Stock Added Succefully

After the user fills the field and press the 'Save' butoon, the server will respond with either a 'Stock with this name already exists' error 5.36, or success message 'Stock Added Successfiuly' in the bottom left of the page and the new stock will be in the bottom of the table 5.37.

5.3.7 Tool's Panel

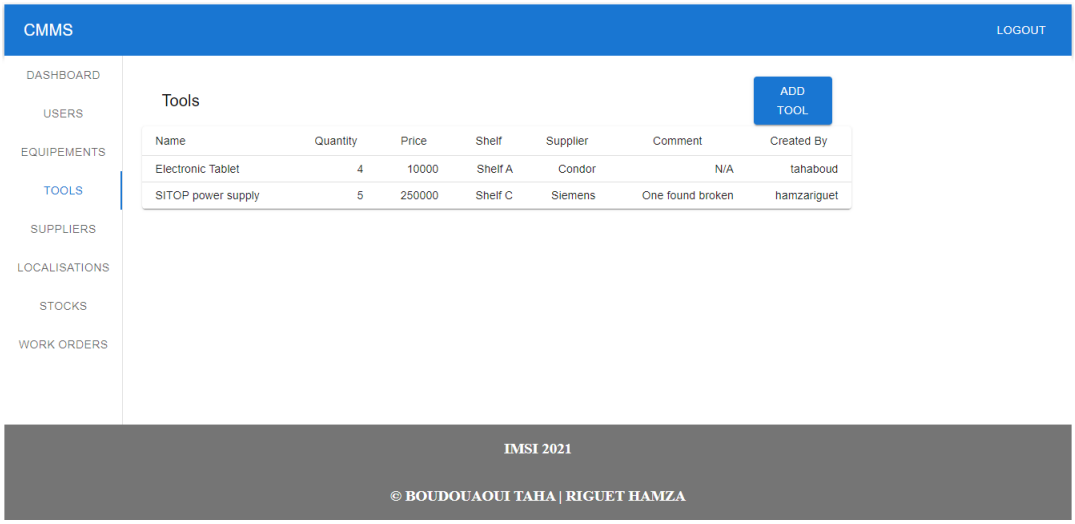

Figure 5.38: Tool Panel

In the tools panel 5.38 we can see all the tools that we have in the stock, described by their name, quantity, price, shelf in the stock, supplier, comment for any additional informations and the user that created the equipement.

And to add a new tool the user clicks on the 'Add Tool' button.

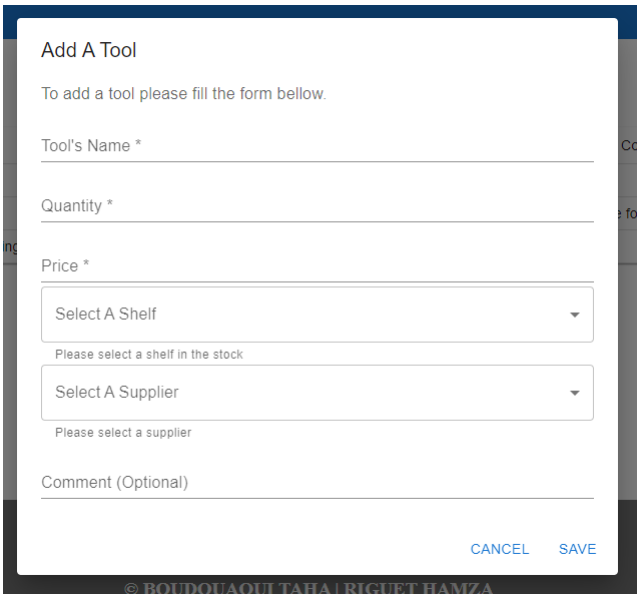

Figure 5.39: Add Tool Dialog

After clicking on the 'Add Tool' button, a dialog will open 5.39 with a form to add a new tool containing five required fields (name, quantity, price, shelf, supplier) and one optional field (comment).

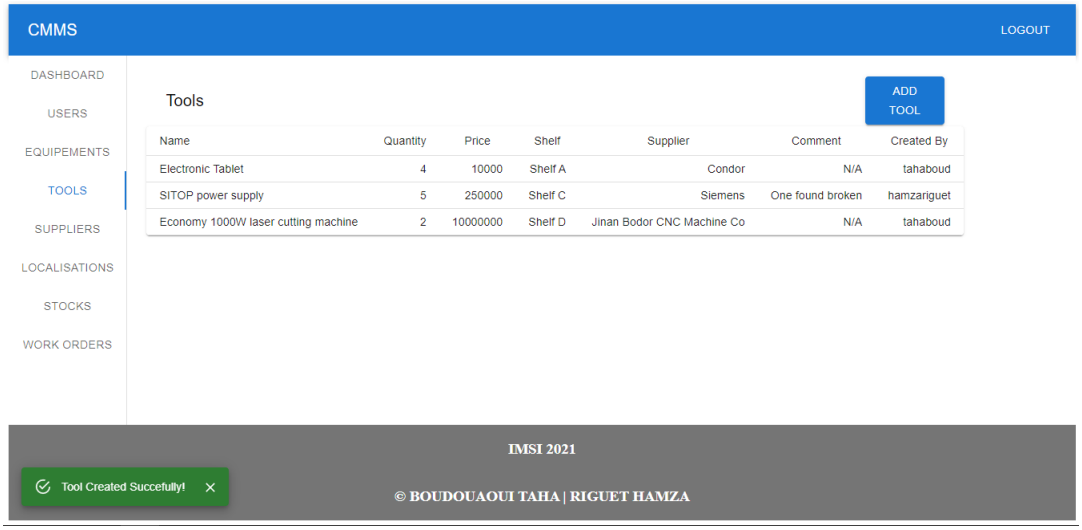

Figure 5.40: Tool Added Successfully

After the user fiils the fields and then clickes on the 'Save button', the server responds with a success message 'Tool Added Succesfully' 5.40.

5.3.8 Work Order Panel

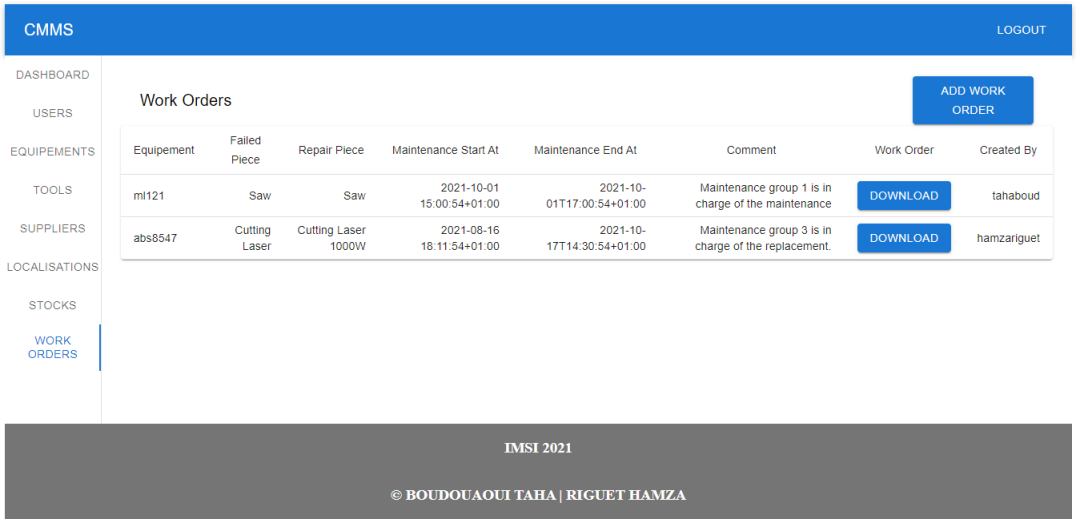

Figure 5.41: Work Order Panel

In the work order panel 5.41 we can see all the work orders that have been created, each work order is described by the equipement, failed piece, repair piece, maintenance start time, maintenance end time, comment for any additional informations and the user that created it, there is also a button to download and print the work order.

And on the top right of the table, there is an 'Add Work Order' button to add a new one.

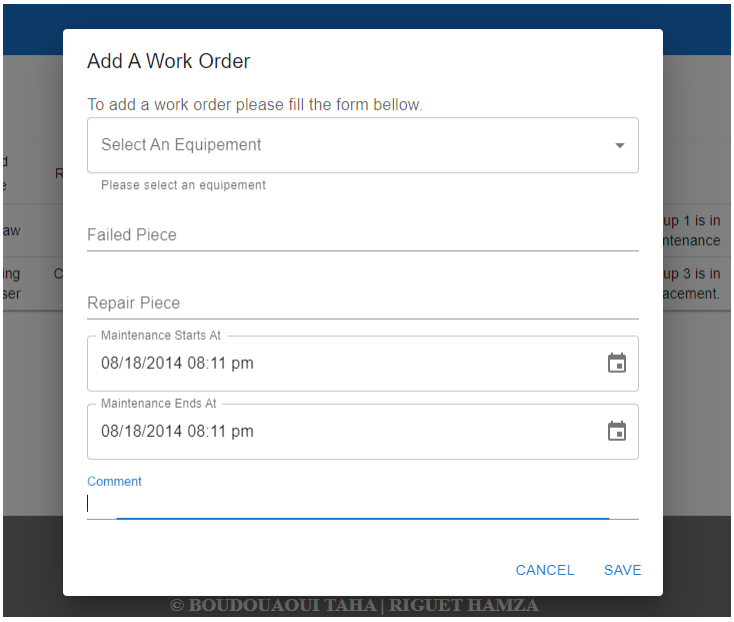

Figure 5.42: Work Order Dialog

After the user clicks on the button, a dialog will pop up 5.42 with a form, it contains an equipement dropdown, a field for the failed piece and a repair piece, a custom user friendly inputs for date and time for the maintenance start and end time and a comment field, and the user clicks on save to create the new work order.

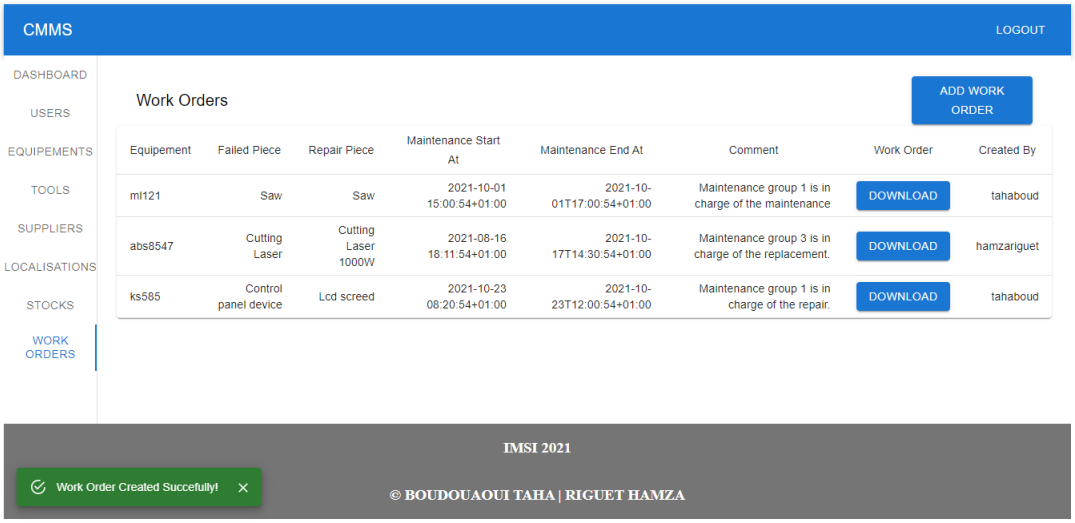

Figure 5.43: Work Order Add Succefully

When the work order is created a response from the server is displayed to the user on the bottom left of the screen 5.43, and then the user clicks on the download butoon to download the pdf version of the work order.

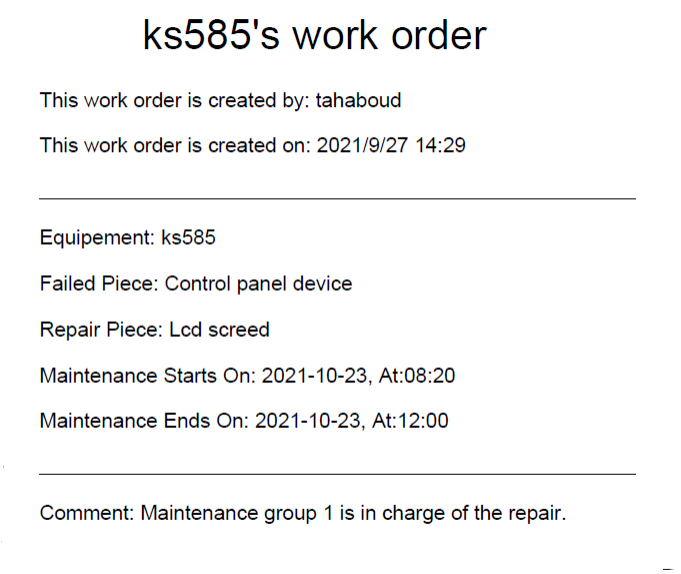

Figure 5.44: Work Order Pdf

The pdf version of the work order5.44 contains the username of the user that created it, the time they created it at, the equipement code, the failed piece, the repair piece, the maintenance start time, the maintenance end time and finaly any comment that was added.

5.4 Conclusion

We have talked about all the programming languages that we used in the creation of our website, its roles and its functionalities, and then we jumped to the website that will simplify all of the maintenance management and organise it, saw all of the functionalities and all the abilities of organizing and simplification

Chapter 6

Conclusion

Managing the upkeep of sophisticated IT systems for businesses is not always simple. Even simple chores like file management might be challenging if they are handled by someone else. Using web-based applications or CMMS (customer relationship management software) to manage these duties can be quite beneficial in a variety of ways. These programs provide a number of advantages.

Using web-based CMMS tools to maintain complex systems can save money by automating duplicate operations and decreasing the need for people for data entry and manual upgrades.

Maintenance management is a crucial component of any successful organization. If maintenance expenditures aren't tracked on a regular basis, they can quickly spiral out of control, whether it's an office, a home, or another type of property investment.

Creating a yearly budget adapted to your specific needs, determining how much work your facility requires, updating the list of activities to ensure they are completed, and assigning the suitable workers to each duty are all part of proper maintenance management.

Maintenance management isn't something you can set and forget. There will always be a need for continuous knowledge updating as new solutions to problems arise.

Bibliography

[1] Sanchit Aggarwal. "Modern web-development using reactjs". In: *International Journal of Recent Research Aspects* 5.1 (2018), pp. 133–137.

[2] *An introduction to web applications architecture*. URL: https://www.open.edu/openlearn/science-maths-technology/introduction-web-applications-architecture/content-section-1.1.

[3] *Asset Panda Software*. URL: https://www.softwareadvice.com/cmms/asset-panda-profile/.

[4] ER R S BANGER. *Client Server Architecture: Diagram, Types, Examples, Components*. URL: https://digitalthinkerhelp.com/what-is-client-server-architecture-diagram-types-examples-components/.

[5] Alex Berson. *Client/server architecture*. McGraw-Hill, Inc., 1996.

[6] Niels Brügger. "Website history and the website as an object of study". In: *New Media & Society* 11.1-2 (2009), pp. 115–132.

[7] Susan C. *What Is HTML? Hypertext Markup Language Basics Explained*. 2021. URL: https://www.hostinger.com/tutorials/what-is-html.

[8] *Class diagrams*. URL: https://java-programming.mooc.fi/part-11/1-class-diagrams.

[9] *Computerized maintenance management system*. URL: https://en.wikipedia.org/wiki/Computerized_maintenance_management_system.

[10] *EZOfficeInventory Software*. URL: https://www.softwareadvice.com/cmms/ezofficeinventory-profile/.

[11] *Fiix Software*. URL: https://www.softwareadvice.com/cmms/fiix-profile/.

[12] Jeff Forcier, Paul Bissex, and Wesley J Chun. *Python web development with Django*. Addison-Wesley Professional, 2008.

[13] Mark Grechanik, Kathryn S McKinley, and Dewayne E Perry. "Recovering and using use-case-diagram-to-source-code traceability links". In: *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*. 2007, pp. 95–104.

[14] *JavaScript*. URL: https://en.wikipedia.org/wiki/JavaScript.

[15] Khairy AH Kobbacy. "Artificial intelligence in maintenance". In: *Complex system maintenance handbook* (2008), pp. 209–231.

[16] Hans J Kohler et al. "Integrating UML diagrams for production control systems". In: *Proceedings of the 2000 International Conference on Software Engineering. ICSE 2000 the New Millennium*. IEEE. 2000, pp. 241–251.

[17] *Lucid Chart*. URL: https://www.lucidchart.com/.

[18] *Maintenance Management Definition*. 2019. URL: https://www.onupkeep.com/answers/asset-management/what-is-maintenance-management.

[19] *Maintenance Management Objectives*. 2020. URL: https://ftmaintenance.com/maintenance-management/8-maintenance-management-objectives/.

[20] Matthew John McGill. "UML Class Diagram Syntax: An Empirical Study of Comprehension." In: (2001).

[21] AB MySQL. *MySQL*. 2001.

[22] Hack Reactor. *What is JavaScript used for?* 2021. URL: https://www.hackreactor.com/blog/what-is-javascript-used-for.

[23] Célia Sagnier. *CMMS: definition and application*. 2019. URL: https://mobility-work.com/blog/cmms-definition-application.

[24] *Sequence diagram*. URL: https://www.ibm.com/docs/SS5JSH_7.5.5/com.ibm.xtools.sequence.doc/topics/cseqd_v.html.

[25] *Smartsheet Software*. URL: https://www.softwareadvice.com/remote-work/smartsheet-profile/.

[26] Taha Sufiyan. *What is ReactJS: Introduction To React and Its Features*. 2021. URL: https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs.

[27] Hippo Content Team. *A.I. in Maintenance Management*. 2018. URL: https://hippocmms.iofficecorp.com/blog/ai-in-maintenance-management.

[28] *UpKeep Software*. URL: https://www.softwareadvice.com/cmms/upkeep-profile/.

[29] *Use case diagram*. URL: https://en.wikipedia.org/wiki/Use_case_diagram.

[30] Guido Van Rossum and Fred L Drake Jr. *Python tutorial*. Vol. 620. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[31] *Website*. URL: https://www.searchmetrics.com/glossary/websites/.

[32] *What Is a Database*. URL: https://www.oracle.com/database/what-is-database/.

[33] *What is CSS?* URL: https://www.tutorialspoint.com/css/what_is_css.htm.

[34] *What is Django?* URL: https://tutorial.djangogirls.org/en/django/.

[35] *What is Python? Executive Summary*. URL: https://www.python.org/doc/essays/blurb/.

[36] Guojun Zhu, Ludo Gelders, and Liliane Pintelon. "Object/objective-oriented maintenance management". In: *Journal of quality in maintenance engineering* (2002).