

DECLARATION

This is to declare that this project titled “**DESIGN AND IMPLEMENTATION OF A BAKERY MANAGEMENT SYSTEM**” is written by **FOBASSO TOUTSOP BOREIL**, a student of the department of **SOFTWARE ENGINEERING** at **AZIMUT SUP**. All statements made and conclusions drawn are outcomes of research works and all borrowed ideas have been acknowledged by means of references quotations

AUTHOR:

FOBASSO TOUTSOP BOREIL

SIGNATURE:

DATE.....

SUPERVISOR

MR. ATOH DENIS

SINATURE.....

DATE.....



CERTIFICATION

This is to certify that this work titled “**DESIGN AND IMPLEMENTATION OF A MANAGEMENT SYSTEM FOR THE MAINTENANCE OF INFORMATION AND TECHNOLOGY EQUIPEMENTS**” has been written and all the research carried and presented by **FOBASSO TOUTSOP BOREIL** under the supervision of **Mr. ATOH DENIS** in achievement for the award of the **HIGHER NATIONAL DIPLOMA (HND)**. It is approved for the contribution to scientific knowledge and literally presented.

ACADEMIC SUPERVISOR

Mr. ATOH DENIS

SIGNATURE

DATE

PROFESSIONAL SUPERVISOR

Mr.

SIGNATURE.....

DATE



DEDICATION

To my family



ACKNOWLEDGMENTS

- ❖ A work of such degree could not have been done and completed without any help and assistance. I address my sincere thanks to the almighty GOD for the strength and his grace in order to achieve this report.
- ❖ Special thanks go to my academic supervisor **Mr. ATOH DENIS** whose directives instructions encouragements and advice helped me to achieve this report.
- ❖ I would also like to thank all the personnel of **CIA CONSULTING** Enterprise for giving me the knowledge and for putting at my disposition the material necessary for me to achieve this report.
- ❖ I also thank particularly my parents and family for all their support both moral and material help, assistance, encouragements to all my brothers and sisters for their support and encouragements to all the persons who took of their time to answer my questionnaires.
- ❖ I would like to express my sincere gratitude to everyone who supported me during my internship. First and foremost, I would like to thank **IT MANAGER Mr. BITYEKI MICHEAL** and the IT staff for their guidance, mentorship, and valuable feedback throughout the internship. Their expertise and insights have been instrumental in shaping my professional growth.
- ❖ Furthermore, I would like to acknowledge the contributions of **Mr. Sonfack Aubert, Mr. AYISSI TEDDY, BERNADETTE** and **GEORGE** for their continuous support and guidance. Their expertise and willingness to share knowledge have been invaluable resources.
- ❖ I am truly grateful for the opportunity to be part of this internship program, and I look forward to applying the skills and knowledge gained in my future endeavors.

Thank you



ABSTRACT



In the world of business, the reference department relies heavily on computer workstations to provide patrons with access to reference sources in CD-ROM and Web formats. Many reference departments also supervise an electronic classroom which is used to provide hands-on instruction. Planning for the hardware, software, and peripherals necessary to provide patrons with access and training must be an ongoing process in order to keep up with rapid technological changes, both in computer hardware and software applications. Through the maintenance of comprehensive records of existing equipment, including the purpose, capabilities and maintenance of each item, information will be readily available for use in planning for computer equipment. In this article, the authors discuss various types of records that should be kept for computer equipment and how the information contained in these records can be applied to ongoing planning and decision making for management and maintenance.



RESUME



Dans le monde des affaires, le département de référence s'appuie fortement sur les postes de travail informatiques pour fournir aux usagers un accès aux sources de référence en formats CD-ROM et Web. De nombreux départements de référence supervisent également une salle de classe électronique qui est utilisée pour fournir une instruction pratique. La planification du matériel, des logiciels et des périphériques nécessaires pour fournir aux usagers un accès et une formation doit être un processus continu afin de suivre l'évolution rapide des technologies, tant en matière de matériel informatique que d'applications logicielles. Grâce à la maintenance de registres complets de l'équipement existant, y compris l'objectif, les capacités et la maintenance de chaque élément, les informations seront facilement disponibles pour être utilisées dans la planification de l'équipement informatique. Dans cet article, les auteurs discutent des différents types de registres qui devraient être conservés pour l'équipement informatique et comment les informations contenues dans ces registres peuvent être appliquées à la planification et à la prise de décision en cours pour la gestion et la maintenance.



TABLE OF CONTENTS

DECLARATIONi

CERTIFICATIONii

DEDICATIONiii

ACKNOWLEDGMENTSiv

ABSTRACTv

RESUMEvi

TABLE OF CONTENTSvii

LIST OF TABLESx

LIST OF FIGURESxi

LIST OF ABBREVIATIONSxii

CHAPTER ONE:1

GENERAL INTRODUCTION1

1

1

1.1.1 Importance of Computer Maintenance:Error! Bookmark not defined.

1.1.2 Types of Computer Maintenance:Error! Bookmark not defined.

1.1.3 Best Practices in Computer Maintenance:Error! Bookmark not defined.

6

7

8

1.4.1 Main research objectives8

1.4.2 Specific research objectives9

1.5 10

1.6 11

1.7.1 HISTORICAL BACKGROUND OF HEVECAM S.AError! Bookmark not defined.

1.7.2 GEOGRAPHICAL DESCRIPTION OF HEVECAMError! Bookmark not defined.



1.7.3 ACTIVITIES OF THE ORGANISATIONError! Bookmark not defined.

1.7.5 PartnersError! Bookmark not defined.

1.7.6 Potential Resources of HEVECAM.SAError! Bookmark not defined.

Error! Bookmark not defined.

Error! Bookmark not defined.

1.9.1 Time scopeError! Bookmark not defined.

1.9.2 Geographical scopeError! Bookmark not defined.

CHAPTER 2 :12

LITTERATURE REVIEW12

12

2.1.1 VISUAL STUDIO CODE12

2.1.2 WEB APPLICATION DEVELOPMENT17

29

Error! Bookmark not defined.

2.3.1 GLPI(Gestionnaire Libre de Parc Informatique)Error! Bookmark not defined.

CHAPTER THREE:32

RESEARCH METHOD A ND INTERNSHIP ACTIVITIES32

INTRODUCTION32

32

3.1.1 Prototyping model33

3.1.2 Justification of choice38

3.1.3 APPLICATION OF THE PROTOTYPING MODEL38

44

3.2.1 Activities carried out in the different departments45

3.2.2 BENEFITS AND ACHIEVEMENTS DERIVED FROM THE INTERNSHIP50

51

INTRODUCTION57

57

58

58

REFERENCES59

APPENDIX61

61



DESIGN AND IMPLEMENTATION OF A BAKERY MANAGEMENT SYSTEM

62

63

65

65

66



LIST OF TABLES

Table 1: Data sheet of HEVECAM SA.....9



LIST OF FIGURES

Figure 1: organizational structure of HEVECAM S.A.	12
Figure 2 :3-tier architecture.....	41
Figure 3: GLPI admin dashboard functionalities.....	41
Figure 4 GLPI admin dashboard functionalities	47
Figure 5 Use case diagrams	51
Figure 6 Activity diagram	52
Figure 7 Sequence diagram.....	53
Figure 8 : Class diagram	54
Figure 9 login page.....	65
Figure 10: Registration page.....	65
Figure 11 Admin dashboard	66
Figure 12 Equipment panel.....	67
Figure 13: log out	67
Figure 14: dataset.....	72
Figure 15	75
Figure 16 Certificate of completion of the internship.....	76
Figure 17 Enterprise evaluation grid	77
Figure 18 prototype 1.....	78
Figure 19 prototype 2.....	78



LIST OF ABBREVIATIONS

ABBREVIATIONS	IMPLICATIONS
AJAX	Asynchronous JavaScript and XML
CDM	Conceptual data model
CSS	Cascading Style sheet
CV	Curriculum Vitae
DB	Database
DBMS	Database Management System
DC	Chart of classes
STD	State-Transition Diagram
GD	General Director
HR	Human resources
HND	Higher National Diploma
IFD	Information Flow Diagram
GLPI	Gestionnaire libre de pack informatique
GUI	Graphical user interface
PC	Personal computer
OOP	Object oriented programming
UML	Unified modeling language



DESIGN AND IMPLEMENTATION OF A BAKERY MANAGEMENT SYSTEM

OS	Operating system
API	Application programing interface
IOT	Internet of things



CHAPTER ONE: GENERAL INTRODUCTION

INTRODUCTION

This report summarizes my internship experience at CIA CONSULTING, a software company which main focus was on implementation of web apps. I worked there as a web developer from July 8, 2024 to September 8, 2024. The purpose of this report is to describe the objectives, activities, challenges, achievements and recommendations of my internship.

During the internship, we were assigned to the information and technology department, where they worked on various development projects, conducted quality assurance tests, and collaborated with cross-functional teams. The primary objectives of the internship included gaining hands-on experience in agile software development methodologies, enhancing coding skills, and understanding the software development life cycle and interacting with workers or personels on daily bases.

This report will dive into the day-to-day activities, challenges faced, accomplishments achieved, and lessons learned during the internship period. By reflecting on the experiences at CIA CONSULTING, we aim to draw valuable insights and practical knowledge that will be beneficial for future career aspirations in the tech industry.

1.1 BACKGROUND STUDY

Software development is a structured process that involves creating, designing, testing, and maintaining software applications and systems. It has evolved significantly over the past few decades, reflecting advancements in both technology and methodologies. As organizations and individuals increasingly rely on technology, the demand for effective, efficient, and secure software continues to grow. Understanding the development process and the underlying principles is crucial for building reliable systems and applications.

1.1.1 Evolution of Software Development

The journey of software development dates back to the mid-20th century, with the emergence of early computing systems. Initially, software was manually written in machine language or assembly code. Early systems were often built for very specific tasks and lacked the complexity of modern applications. Over time, as computing technology advanced, the need for structured software development processes became more apparent.

In the 1960s and 1970s, software engineering emerged as a discipline. The term "software engineering" was coined to describe the systematic approach needed to create large-scale software systems. One of the earliest methodologies to address this was the Waterfall Model, which follows a linear, step-by-step approach. However, as systems grew more complex and requirements changed more frequently, this rigid methodology proved inefficient.

1.1.2 Rise of Agile Methodologies

In the 1990s, a shift toward more flexible and iterative approaches in software development took place. Agile methodologies, emphasizing adaptability, collaboration, and incremental progress, gained prominence. The Agile Manifesto, published in 2001, outlined key principles for software development, such as delivering working software frequently, responding to change, and valuing individuals and interactions over processes and tools. Scrum, Extreme Programming (XP), and Kanban are some of the widely adopted frameworks that focus on iterative cycles, short development sprints, and continuous improvement.

Agile has become the standard in many industries due to its ability to manage uncertainty and rapidly changing requirements, making it particularly suitable for fast-paced, dynamic environments. The focus on collaboration between developers, stakeholders, and customers ensures that the software developed aligns closely with user needs.

1.1.3 Key Phases in Software Development

Regardless of the methodology, software development generally involves several phases, including:

- **Requirements Gathering and Analysis:** Understanding the needs of the user or client is essential. This phase involves defining the functional and non-functional requirements, including performance, security, and scalability.
- **Design:** This phase includes creating architectural designs and selecting appropriate technologies to meet the specified requirements. The system design is often broken down into high-level design (system architecture) and low-level design (detailed component design).
- **Implementation:** The development team writes the code based on the design specifications. This phase may involve programming in various languages such as Java, Python, or C++ and using integrated development environments (IDEs) or version control systems (e.g., Git).
- **Testing:** Comprehensive testing ensures that the software is free from defects and functions as intended. Testing can include unit testing, integration testing, system testing, and acceptance testing. Automation tools and frameworks are often used to improve efficiency.
- **Deployment and Maintenance:** After testing, the software is deployed to the production environment, where it is used by the end users. Continuous monitoring, bug fixing, and performance improvements are part of ongoing maintenance.

1.1.4 Software Development Tools and Technologies

Modern software development is supported by a wide range of tools and technologies that enhance productivity, quality, and collaboration:

- **Version Control Systems (VCS):** Git and platforms like GitHub or GitLab allow developers to track code changes, collaborate with other developers, and maintain code integrity.
- **Integrated Development Environments (IDEs):** Tools like Visual Studio, IntelliJ IDEA, and Eclipse provide code editing, debugging, and testing functionalities in one place.
- **Continuous Integration/Continuous Deployment (CI/CD):** These practices automate the building, testing, and deployment processes, ensuring faster and more reliable software delivery.
- **Frameworks and Libraries:** Software frameworks (e.g., React, Laravel, Django, Spring) provide reusable code that accelerates development and ensures consistency.
- **Cloud Computing and DevOps:** Cloud platforms (e.g., AWS, Azure, Google Cloud) provide scalable infrastructure, while DevOps practices focus on automating and integrating the processes between software development and IT operations.

1.1.4 Challenges in Software Development

Software development, while essential, is not without its challenges. These include:

- **Complexity:** As software systems become more sophisticated, managing complexity becomes more difficult. Ensuring maintainability, scalability, and performance in large systems can be a major challenge.
- **Security:** Ensuring the security of software systems is a critical concern, especially in an era of rising cyber threats. Incorporating security measures into the development lifecycle (DevSecOps) is necessary to protect user data and maintain trust.

- **Changing Requirements:** Software projects often suffer from scope creep, where requirements change frequently or are poorly defined. Agile methodologies help manage this but still require strong communication and clear expectations.
- **Resource Management:** Managing resources, including developers, time, and budget, is vital for delivering a project on schedule. Delays, mismanagement, and underestimation can lead to project failure.

1.1.6 Impact of Software Development on Society

The impact of software development on society is profound. Software is the backbone of numerous industries such as healthcare, finance, education, and entertainment. It powers everything from personal devices like smartphones to critical systems used in transportation and communications. The digital transformation enabled by software development has led to significant economic growth, increased efficiency, and more personalized user experiences.

However, software development also raises ethical questions. Issues such as data privacy, algorithmic biases, and the environmental impact of software and hardware systems are important concerns. Developers are increasingly called to consider the societal and ethical implications of the technologies they create.

Conclusion

The field of software development is continuously evolving to meet the demands of an increasingly digital world. From its early days of simple programs to the complex, global-scale systems of today, software development plays a critical role in shaping modern society. While challenges exist, advancements in methodologies, tools, and technologies continue to drive innovation and ensure that software can meet the needs of users and businesses alike. The future of software development is intertwined with emerging trends such as artificial intelligence, machine learning, blockchain, and quantum computing, which will shape the next generation of applications and systems.

1.2 PROBLEM STATEMENT

Here are some problem encountered in the management of a bakery in the traditional way

1.2.1 Inventory Management Challenges

The bakery faces challenges in tracking inventory, leading to overstocking or understocking of ingredients. This results in wastage, increased costs, and disruptions in production schedules.

1.2.2 Order Management Issues

Manual order handling often results in miscommunication, delayed order processing, and errors in fulfilling customer requirements, affecting customer satisfaction and retention.

1.2.3 Inefficient Staff Scheduling

The bakery struggles to manage staff schedules effectively, leading to overstaffing during low-demand hours and understaffing during peak hours, causing inefficiencies and reduced productivity.

1.2.4 Lack of Real-Time Sales Analytics

The absence of a real-time sales tracking system prevents the bakery from identifying best-selling items, understanding customer preferences, and making data-driven decisions to optimize offerings.

1.2.5 Poor Customer Relationship Management

The bakery lacks a system to maintain customer data, track loyalty, and manage feedback, leading to missed opportunities for personalized marketing and improved customer engagement.

1.2.6 Payment Processing Delays

The current manual or semi-digital payment system causes delays during busy hours, leading to long queues, frustrated customers, and lost sales opportunities.

1.2.7 Ineffective Waste Management

The bakery does not have a system to monitor and minimize waste effectively, which contributes to higher operational costs and environmental concerns

1.2.8 Difficulty in Menu Updates

The bakery faces challenges in updating its menu for seasonal items, offers, or price changes due to the lack of a centralized, easily modifiable system.

1.2.9 Limited Online Presence

The bakery lacks an integrated online ordering and delivery platform, resulting in missed opportunities to cater to the growing demand for online services.

1.2.10 Compliance with Food Safety Standards

Ensuring compliance with food safety regulations is difficult without a proper tracking and reporting mechanism, which could expose the bakery to regulatory risks.

1.3 RESEARCH QUESTION

1. How can a digital bakery management system improve operational efficiency and customer satisfaction in a small-to-medium-sized bakery?
2. What impact does an automated inventory management system have on reducing wastage and ensuring the availability of essential ingredients in a bakery?
3. How can integrating an online ordering platform with a bakery management system enhance customer convenience and loyalty?
4. How can real-time sales analytics help a bakery identify trends and optimize its product offerings to maximize profitability?
5. What role can a digital management system play in minimizing waste and promoting sustainability in a bakery's operations?
6. How can an automated staff scheduling feature improve labor utilization and reduce inefficiencies in a bakery?
7. Can a bakery management system help streamline compliance with food safety standards and reduce the risk of violations?

8. How does the integration of online ordering and in-store operations in a bakery management system impact overall business performance?
9. How does the implementation of a digital payment system in a bakery management system influence transaction speed and customer satisfaction?
11. What design features of a bakery management system can ensure scalability and adaptability to meet the evolving needs of a growing bakery business?

1.4 OBJECTIVES OF THE STUDY

1.4.1 Main research objectives

During this internship period our main research objectives where to ;

Objective 1: the design and implementation of a user-friendly interface for inventory management employees tracking and real-time analytics.

Objective 2: To implement a bakery management system that streamlines core operations and enhances overall efficiency.

Objective 3: To improve inventory management and reduce wastage through an automated tracking system.

Objective 4: To enhance customer satisfaction by simplifying order management and integrating online ordering options.

Objective 5: Develop reporting tools that generate detailed analytics, performance metrics, and historical data on sails informed decision-making, resource planning, and identification of trends for process optimization.

Objective 6: To optimize staff scheduling and resource allocation for improved productivity.

Objective 7: To ensure the scalability and adaptability of the system for future business growth and changing requirements.

Objective 8: Implement real-time updates and notification systems within the software to alert relevant stakeholders about task assignments, completions, delays, or critical issues, ensuring timely communication and decision-making.

Objective 9: Conduct extensive user testing sessions with maintenance personnel, supervisors, and administrators to gather feedback, identify usability issues, and iteratively improve the software based on user perspectives and suggestions.

1.4.2 Specific research objectives

Objective 1: Design and implementation of an inventory management module that tracks stock levels and minimizes ingredient wastage.

Objective 2: Integration of a real-time sales analytics tool for monitoring product performance and customer preferences.

Objective 3: To develop a customer order management feature that ensures accuracy and reduces processing errors.

Objective 4: Implement mobile compatibility features that enable clients, order online, receive push notifications, and update task status in real time while in the field.

Objective 5: Creation of a staff scheduling system that optimizes workforce allocation based on demand patterns.

Objective 6: Ensure compliance with food safety standards through automated documentation and monitoring.

Objective 7: Evaluation of the user experience of the bakery management system among staff and customers.

1.5 SIGNIFICANCE OF STUDY

1. Operational Efficiency: The system streamlines day-to-day operations, such as inventory management, order processing, and staff scheduling, reducing manual errors and saving time.

2. Cost Savings: Automated inventory tracking helps minimize wastage and optimize resource utilization, leading to lower operational costs.

3. Data-Driven Decisions: Real-time sales analytics enable the bakery to identify trends, forecast demand, and optimize product offerings for higher profitability.

4. Improved Customer Experience: Features like online ordering, faster payments, and personalized loyalty programs enhance customer satisfaction and retention.

5. Simplified Workflows: Automation of repetitive tasks reduces workload, allowing staff to focus on customer service and product quality.

6. Optimized Scheduling: A staff scheduling feature ensures balanced workloads, leading to higher productivity and job satisfaction.

7. Convenience: Online ordering and faster checkout processes improve the overall customer experience.

8. Personalization: Enhanced customer relationship management allows for personalized offers and better customer engagement.

9. Foundation for Innovation: This study serves as a reference for developing or improving management systems in other businesses, particularly in the food and beverage industry.

10. Scalability Insights: Insights into scalability and adaptability can guide similar projects for larger businesses or franchises.

11. Adoption of Digital Tools: The study highlights the potential of digital systems in improving small and medium-sized enterprises (SMEs), encouraging more businesses to adopt technology for growth.

1.6 LIMITATIONS

it is important to consider the limitations or challenges that may arise during the development and implementation process. Here are a few potential limitations that we considered:

- 1. Limited Scope:** The system is designed specifically for bakery operations, which may limit its applicability to other types of businesses without significant modifications.
- 2. Resource Constraints:** The development and implementation process may face challenges due to limited financial resources, hardware availability, or technical expertise.
- 3. Time Constraints:** The time allocated for the development and testing phases of the project may not be sufficient to explore all potential features or address all possible use cases.
- 4. User Adoption Challenges:** Employees or stakeholders unfamiliar with digital tools may face a learning curve, which could delay full adoption of the system.
- 5. Data Dependence:** The system's effectiveness depends on accurate and timely input of data, which may not always be ensured due to human error or neglect.
- 6. Scalability Issues:** While designed with scalability in mind, the system may require further enhancements to support a significantly larger customer base or additional bakery branches.
- 7. Internet Dependency:** Features like online ordering and cloud-based data storage rely on stable internet connections, which may not be consistently available in all locations.
- 8. Limited Real-World Testing:** The study might not account for all real-world scenarios, particularly unexpected operational challenges or edge cases during peak business hours.
- 9. Security Concerns:** If adequate security measures are not implemented, sensitive customer and business data might be vulnerable to breaches or misuse.
- 10. Regulatory Compliance:** The system may not fully address all region-specific regulatory requirements, such as food safety standards or data privacy laws, without additional customization.
- 11. Data security and privacy:** Recording sales activities involves handling sensitive data. Ensuring data security, privacy, and compliance with relevant regulations (e.g., GDPR) can be a significant challenge that needs to be addressed.

CHAPTER 2 : LITTERATURE REVIEW

2.1 THEORITICAL REVIEW

Computer science, as a rapidly evolving field at the forefront of technological innovation, plays a pivotal role in shaping the digital landscape and driving advancements in various domains. This literature review aims to explore the diverse spectrum of research, theories, methodologies, and applications within computer science, spanning from foundational principles in algorithms and data structures to cutting-edge developments in artificial intelligence, cyber security, and software engineering. By synthesizing existing literature, analyzing key trends, and identifying research gaps, this review seeks to offer a comprehensive overview of the current state of computer science research, highlighting emerging paradigms, challenges, and opportunities for future exploration. In this chapter we are going to talk about some theoretical concept related to our work. We are also going to analyses some existing solutions in order to bring out their strength and weakness.

In this section we are going to discuss about some concepts and technologies we will use in our work and their different advantages and disadvantages.

2.1.1 VISUAL STUDIO CODE

Visual Studio is an Integrated Development Environment (IDE) developed by Microsoft to develop Desktop applications, GUI (Graphical User Interface), console, web applications, mobile applications, cloud, and web services, etc. With the help of this IDE, you can create managed code as well as native code. It uses the various platforms of Microsoft software development software like Windows store,

Microsoft Silverlight, and Windows API. It is not a language-specific IDE as you can use this to write code in C#, C++, VB (Visual Basic), Python, JavaScript, and many more languages. It provides support for 36 different programming languages. It is available for Windows as well as for macOS. (<https://www.geeksforgeeks.org/introduction-to-visual-studio/>, 2024)

Evolution of Visual Studio

The first version of VS (Visual Studio) was released in 1997, named as Visual Studio 97 having version number 5.0. The latest version of Visual Studio is 15.0 which was released on March 7, 2017. It is also termed as Visual Studio 2017. The supported .Net Framework Versions in latest Visual Studio is 3.5 to 4.7. Java was supported in old versions of Visual Studio but in the latest version doesn't provide any support for Java language.

Visual Studio Editions

There are 3 editions of Microsoft Visual Studio as follows:

1. Community

It is a free version which is announced in 2014. All other editions are paid. This contains the features similar to Professional edition. Using this edition, any individual developer can develop their own free or paid apps like .Net applications, Web applications and many more. In an enterprise organization, this edition has some limitations. For example, if your organization have more than 250 PCs and having annual revenue greater than \$1 Million (US Dollars) then you are not permitted to use this edition. In a non-enterprise organization, up to five users can use this edition. Its main purpose is to provide the Ecosystem (Access to thousands of extensions) and Languages (You can code in C#, VB, F#, C++, HTML, JavaScript, Python, etc.) support.

2. Professional

It is the commercial edition of Visual Studio. It comes in Visual Studio 2010 and later versions. It provides the support for XML and XSLT editing and includes the tool like Server Explorer and integration with Microsoft SQL Server. Microsoft provides a free trial of this edition and after the trial period, the user has to pay to continue using it. Its main purpose is to provide Flexibility (Professional developer tools for building any application type), Productivity (Powerful features such as Code Lens

improve your team's productivity), Collaboration (Agile project planning tools, charts, etc.) and Subscriber benefits like Microsoft software, plus Azure, Plural sight, etc.

3. Enterprise

It is an integrated, end to end solution for teams of any size with the demanding quality and scale needs. Microsoft provides a 90-days free trial of this edition and after the trial period, the user has to pay to continue using it. The main benefit of this edition is that it is highly scalable and deliver high-quality software.

Getting Started with Visual Studio 2017

- First, you have to download and install the Visual Studio. For that, you can refer to Downloading and Installing Visual Studio 2017. Don't forget to select the .NET core workload during the installation of VS 2017. If you forget then you have to modify the installation.

- You can see a number of tool windows when you will open the Visual Studio and start writing your first program as follows:

Code Editor: Where the user will write code.

- Output Window: Here the Visual Studio shows the outputs, compiler warnings, error messages and debugging information.

- Solution Explorer: It shows the files on which the user is currently working.

- Properties: It will give additional information and context about the selected parts of the current project.

- A user can also add windows as per requirement by choosing them from View menu. In Visual Studio the tool windows are customizable as a user can add more windows, remove the existing open one or can move windows around to best suit.

- Various Menus in Visual Studio: A user can find a lot of menus on the top screen of Visual Studio as shown below

- Create, Open and save projects commands are contained by File menu.

- Searching, Modifying, Refactoring code commands are contained by the Edit menu.
- View Menu is used to open the additional tool windows in Visual Studio.
- Project menu is used to add some files and dependencies in the project.
- To change the settings, add functionality to Visual Studio via extensions, and access various Visual Studio tools can be used by using Tools menu.
- The below menu is known as the toolbar which provide the quick access to the most frequently used commands. You can add and remove the commands by going to View → Customize

ADVANTAGES OF USING VISUAL STUDIO IDE

- A full-featured programming platform for several operating systems, the web, and the cloud, Visual Studio IDE is available. Users can easily browse the UI so they can write their code quickly and precisely.
- To help developers quickly identify potential errors in the code, Visual Studio offers a robust debugging tool.
- Developers can host their application on the server with confidence because they have eliminated anything that could lead to performance issues.
- No matter what programming language developers are using, users of Visual Studio can get live coding support. For faster development, the Platform offers an autocomplete option. The built-in intelligent system offers descriptions and tips for APIs.
- Through Visual Studio IDE you can easily collab with your teammates in a same project. This IDE helps the developers to share, push and pull their code with their teammates.
- Every user of Visual Studio has the ability to customize it. They have the option to add features based on their needs. For example, they can download add-ons and install extensions in their IDE. Even programmers can submit their own extensions.

DISADVANTAGES OF USING VISUAL STUDIO IDE

- **Resource Consumption:** Since VS Code is built on top of Chromium, it can consume more memory and CPU resources compared to lightweight text editors. Some users may find it slower or resource-intensive.
- **Telemetry:** By default, VS Code collects telemetry data to improve the product. While this helps the development team, some users may be concerned about privacy.

Telemetry: By default, VS Code collects telemetry data to improve the product. While this helps the development team, some users may be concerned about privacy.

Lack of Integrated Help: Unlike some IDEs, VS Code lacks an integrated help system that directly links to tutorials, videos, or documentation. Users need to rely on external resources.

Limited Built-in Features: While extensions enhance functionality, some users might miss certain features commonly found in full IDEs, such as built-in database tools or advanced project management.

Initial Learning Curve: The extensive range of features can be overwhelming for beginners. However, once users become familiar with the tool, it becomes more intuitive.

No Native Cloud Settings Sync: Although there are workarounds, VS Code does not natively provide cloud-based

Note:

- Support for different programming languages in Visual Studio is added by using a special Package which is known as Language Service.
- When you will install the Visual Studio then the functionality which is coded as VSPackage will be available as Service.
- Visual Studio IDE provides the three different types of services known as SVsSolution, SVsUIShell, and SVsShell.
- SVsSolution service is used to provide the functionality to enumerate solutions and projects in Visual Studio.

- SVsUIShell service is used to provide User Interface functionality like toolbars, tabs etc.
- SvsShell service is used to deal with the registration of VSPackages.

2.1.2 WEB APPLICATION DEVELOPMENT

Web application development is the creation of application programs that reside on remote servers and are delivered to the user's device over the Internet. A web application (web app) does not need to be downloaded and is instead accessed through a network. An end user can access a web

Application through a web browser such as Google Chrome, Brave, Safari, or Mozilla Firefox. A majority of web applications can be written in JavaScript, Cascading Style Sheets (CSS), and HTML5. Web application development will typically have a short development life-cycle lead by a small development team. Front-end development for web applications is accomplished through client-side programming. Client refers to a computer application such as a web browser. Client-side programming will typically utilize HTML, CSS and JavaScript. HTML programming will instruct a browser how to display the on-screen content of web pages, while CSS keeps displayed information in the correct format. JavaScript will run JavaScript code on a web page, making some of the content interactive. Server-side programming powers the client-side programming and is used to create the scripts that web applications use. Scripts can be written in multiple scripting languages such as Ruby, Java and Python. Server-side scripting will create a custom interface for the end-user and will hide the source code that makes up the interface (Web application development, 2023). A database such as MySQL, SQLITE3 or Mongo DB can be used to store data in web application development. There are many tools and programming languages that can be used to create dynamic and interactive web pages. PYTHON, PHP, ASP.NET and JSP are the most popular programming tools for develop web pages.

a. PHP

PHP (Hypertext Preprocessor) is a popular open-source server-side scripting language designed specifically for web development. It is embedded within HTML, allowing developers to create dynamic and interactive web pages efficiently. PHP excels in handling server-side tasks like managing databases, processing forms, and generating dynamic content.

Some of the companies that use PHP as their technology stacks include:

- 1. Facebook** One of the largest social media platforms in the world. Facebook initially used PHP extensively to build its dynamic web interface. Although it now uses a custom version called **Hack**, PHP remains the foundation of its development.
- 2. WordPress:** The most popular Content Management System (CMS), powering over 40% of websites globally. PHP forms the core of WordPress, enabling the creation and management of blogs, e-commerce sites, and corporate websites with themes and plugins.
- 4. Wikipedia:** The world's largest online encyclopedia. Wikipedia uses PHP to manage its extensive content, ensure fast page loads, and handle its massive global traffic.
- 5. Slack:** A leading team collaboration and messaging platform. PHP is used in Slack's back-end infrastructure for certain operations,

A. ADVANTAGES OF PHP

Despite the fact that PHP has been around for a long time and is well known to everyone, there is often a gap in understanding its advantages over other programming languages, especially when used in web development. Let's list some of these advantages.

- 1. Open Source and Free:** PHP is open-source and available for free, making it cost-effective for businesses and developers. The large community contributes to constant updates, resources, and support.
- 2. Platform Independence:** PHP runs on all major operating systems, including Windows, Linux, and macOS. Its cross-platform compatibility ensures seamless integration with various systems and web servers like Apache and Nginx.
- 3. Ease of Learning and Use:** PHP has a simple and intuitive syntax, making it beginner-friendly. Developers with basic programming knowledge can quickly learn and start building applications with PHP.
- 4. Rich Ecosystem of Frameworks:** PHP offers a variety of frameworks like Laravel, Symfony, CodeIgniter, and Yii. These frameworks simplify development by providing built-in features for authentication, routing, and database management.

5. **Extensive Database Support:** PHP supports a wide range of databases, including MySQL, PostgreSQL, SQLite, Oracle, and MongoDB. This versatility allows developers to choose the best database for their application.
6. **Large Community and Resources:** PHP boasts a vast community of developers, resulting in abundant resources, tutorials, forums, and libraries. This makes troubleshooting and learning more accessible.
7. **Scalability:** PHP supports building scalable web applications, from small blogs to enterprise-level systems like Facebook and Wikipedia. Frameworks and tools further enhance its scalability.
8. **Fast Development and Deployment:** PHP enables rapid application development with its built-in functions, extensive libraries, and frameworks. It is especially efficient for small to medium-scale applications.
9. **Integration with Other Technologies:** PHP integrates seamlessly with various technologies, including HTML, CSS, JavaScript, and XML. It also works well with APIs, web services, and CMS platforms.
10. **High Performance:** With modern versions (PHP 7 and PHP 8), PHP has significantly improved in performance. Features like Just-In-Time (JIT) compilation enhance execution speed, making it suitable for high-traffic applications.
11. **Security Features:** PHP includes built-in security features, such as input validation, encryption, and mechanisms to prevent SQL injection, XSS, and CSRF attacks. Frameworks further enhance security with pre-configured tools.
12. **Widespread Hosting Support:** Most web hosting providers offer PHP support out of the box, making it easier to deploy applications without special configurations.

1. Web Development Using PHP Frameworks

Another good thing about web development using PHP is that it has many frameworks that simplify the development process which provide pre-built tools, libraries, and structures to streamline the development process, enhance productivity, and promote best practices like MVC (Model-View-Controller) architecture. Depending on what you're doing, you may need different frameworks. Let's take a look at the most well-known PHP frameworks

1. Laravel

- Elegant syntax and developer-friendly features.
- Built-in tools for routing, authentication, caching, and database migrations.
- Offers Eloquent ORM for seamless database management.
- Includes Blade, a lightweight templating engine.
- Supports task scheduling and queues for complex workflows.
- Ideal for: Large-scale applications and projects requiring robust features.

2. Symfony

- Highly flexible and modular, with reusable components.
- Implements a strong focus on performance and scalability.
- Provides comprehensive documentation and a vibrant community.
- Supports third-party integrations and API development.
- Includes Twig, a powerful templating engine.
- Ideal for: Enterprise-level applications and developers who need customization.

3. CodeIgniter

- Lightweight and fast with minimal configuration.
- Suitable for small-scale applications and projects with tight deadlines.
- Provides simple tools for database management and file uploading.
- Lacks built-in ORM, but supports raw SQL queries.
- Ideal for: Beginners and small-to-medium-sized applications.

4. CakePHP

- Emphasizes convention over configuration for rapid development.
- Features built-in validation, security tools, and CRUD scaffolding.
- Provides powerful tools for unit testing and database management.
- Supports backward compatibility across versions.
- Ideal for: Developers seeking a structured and consistent framework.

5. Zend Framework (Laminas)

- Now maintained under the Laminas Project.
- Offers enterprise-grade solutions with a modular approach.
- Includes tools for MVC implementation, authentication, and caching.
- Supports Restful API development and cloud integration.
- Ideal for: Enterprise applications with complex requirements.

6. Phalcon

- Written in C and delivered as a PHP extension, making it exceptionally fast.
- Low memory usage and high performance.
- Includes tools for ORM, templating, and RESTful APIs.
- Provides a flexible architecture for high-speed applications.
- Ideal for: Performance-critical projects.

7. Yii Framework

- Focuses on simplicity, efficiency, and extensibility.
- Features Gii, a powerful code generator for CRUD operations.
- Provides built-in tools for caching, authentication, and role-based access control (RBAC).
- Good support for RESTful APIs and AJAX integration.
- Ideal for: Rapid development of web applications with complex workflows.

8. Slim Framework

- A minimalist framework for building simple yet powerful APIs and microservices.
- Offers a lightweight design with a focus on routing and middleware.
- Easily extensible with third-party libraries.
- Ideal for: Building lightweight APIs and microservices.

9. FuelPHP

- Implements the HMVC (Hierarchical Model-View-Controller) architecture for improved modularity.

- Offers strong security features, including input filtering and CSRF protection.
- Supports RESTful APIs and flexible templating.
- Ideal for: Developers who prefer HMVC over traditional MVC.

10. PHPixie

- Lightweight and easy to set up.
- Focuses on high performance and simplicity.
- Built for HMVC architecture and includes ORM for database management.
- Ideal for: Beginners or developers working on simple web applications.

2. DISADVANTAGES OF PHP

1. Inconsistent Syntax: PHP's syntax has evolved over the years, leading to inconsistencies that can confuse developers. Leading to errors and increased time spent learning or debugging code.

2. Security Vulnerabilities: While PHP provides security features, its ease of use and prevalence make it a common target for attackers. Applications may become vulnerable to SQL injection, XSS, and other attacks if developers do not follow best practices.

3. Performance Limitations: PHP is interpreted at runtime, which may lead to slower performance compared to compiled languages like Java or C++. While modern versions (PHP 7 and PHP 8) have improved performance, it might not be the best choice for extremely high-performance applications.

4. Lack of Modular Design: Older PHP projects often suffer from a lack of modularity and separation of concerns. Maintaining and scaling such applications can be difficult without frameworks or proper coding practices.

5. Limited Suitability for Modern Needs: PHP was originally designed for server-side scripting and may not cater as effectively to modern, real-time, or microservices architectures. Developers may prefer other languages like Node.js for building modern, real-time applications.

6. **Weak Typing:** PHP's dynamic typing can lead to unexpected behaviors and subtle bugs, especially in large, complex applications. Debugging can become cumbersome, and enforcing type safety may require additional effort.
7. **Over-Reliance on Community Contributions:** PHP's extensive ecosystem depends heavily on third-party libraries and frameworks. Poorly maintained libraries or lack of official support can lead to compatibility and security issues.
8. **Perception as Outdated:** PHP is often viewed as an older technology compared to newer languages like Python, Ruby, or JavaScript. This perception might make it harder to attract developers to PHP projects.
9. **Poor Error Handling in Older Versions:** Error handling in PHP (prior to modern versions) was not as robust or developer-friendly. Debugging and managing errors were more difficult, though this has improved with newer versions.
10. **Tight Coupling with Web Development:** PHP is primarily designed for web development, making it less suitable for non-web applications like standalone software or mobile apps. Developers seeking to create versatile or hybrid applications may choose other languages.

B. JAVASCRIPT

JavaScript is a high-level, versatile, and interpreted programming language primarily used for front-end web development. It allows developers to create interactive and dynamic user interfaces by manipulating the DOM (Document Object Model) and handling user events in real time. JavaScript runs directly in web browsers, making it essential for modern web applications. With the advent of frameworks and libraries like React, Angular, and Vue.js, JavaScript has become even more powerful and indispensable for creating rich, responsive web experiences.

A. ADVANTAGES OF JAVASCRIPT

1. **Client-Side Execution:** JavaScript code runs directly in the user's browser, reducing server load and enabling faster interaction. This makes applications more responsive and capable of providing real-time updates.

2. **Versatility and Compatibility:** JavaScript is supported by all major web browsers and can be used with other technologies like HTML and CSS. It ensures cross-platform compatibility and seamless integration into any web development project.
3. **Rich User Interfaces:** Enables the creation of interactive features such as sliders, drag-and-drop functionality, and animations. Enhances user experience and engagement.
4. **Extensive Ecosystem:** A vast number of frameworks and libraries like React, Angular, Vue.js, and jQuery simplify development. Increases productivity and allows developers to focus on application logic instead of repetitive tasks.
5. **Asynchronous Programming:** JavaScript supports asynchronous operations using tools like Promises and async/await. Improves performance by allowing non-blocking operations, especially for tasks like API calls and file processing.
6. **Growing Community and Resources:** JavaScript has one of the largest developer communities, providing extensive tutorials, forums, and libraries. Learning and troubleshooting become easier, especially for beginners.
7. **Server-Side Capabilities:** With Node.js, JavaScript can also be used for server-side programming. Developers can use JavaScript for full-stack development, reducing the need to learn multiple languages.

B. DISADVANTAGES JAVASCRIPT

1. **Security Concerns:** JavaScript code is executed on the client side, making it susceptible to vulnerabilities like cross-site scripting (XSS). Developers need to implement strict security measures to protect applications.
2. **Browser Dependency:** JavaScript execution depends on the browser's engine (e.g., V8 for Chrome, SpiderMonkey for Firefox). Inconsistent behavior may occur across different browsers, requiring additional testing and compatibility adjustments.

3. Debugging Complexity: Errors in JavaScript, such as those related to asynchronous programming or browser-specific quirks, can be challenging to debug. Debugging can be time-consuming, especially for complex applications.

4. Performance Limitations: Being an interpreted language, JavaScript may be slower than compiled languages like C++ or Java. It may not be ideal for computationally intensive tasks, although modern engines have improved performance significantly.

5. Overhead from Frameworks: The extensive use of libraries and frameworks can introduce unnecessary bloat and reduce performance if not managed carefully. Applications may become slow, especially for users with limited resources or slower internet connections.

6. Lack of Static Typing: JavaScript is dynamically typed, leading to potential runtime errors due to type mismatches. This can make code harder to maintain, though TypeScript (a superset of JavaScript) mitigates this issue.

7. Fast-Paced Evolution: The JavaScript ecosystem evolves rapidly, with frequent updates and new frameworks. Developers may struggle to keep up with the latest tools and best practices.

Popular JavaScript Frameworks and Their Particularities

JavaScript frameworks simplify front-end development by providing pre-built components, tools, and structures to create dynamic, scalable, and maintainable applications. Here's an overview of the most popular JavaScript frameworks and their unique features:

1. React developed by META

- Component-based architecture for building reusable UI elements.
- Uses a virtual DOM for efficient rendering and updates.
- Offers React Hooks for managing state and life-cycle.
- Extensive ecosystem with tools like Redux (state management) and Next.js (server-side rendering).
- Ideal For Single-page applications (SPAs) and dynamic user interfaces.

2. Angular developed by GOOGLE

- Implements a two-way data binding mechanism for automatic synchronization

between the model and view.

- Uses TypeScript for enhanced code quality and scalability.
- Provides built-in tools for dependency injection, routing, and form validation.
- Supports reactive programming with RxJS.

3. Vue.js

- Lightweight and easy to learn.
- Offers a reactive data-binding system.
- Combines the best features of Angular and React, such as templates and a virtual DOM.
- Highly flexible and can be integrated into existing projects incrementally.

4. Svelte

- No virtual DOM—components are compiled into efficient vanilla JavaScript at build time.
- Extremely fast with reduced runtime overhead.
- Simple syntax and minimal boilerplate code.
- Smaller bundle sizes compared to React or Angular.

5. Next.js developed by VERCEL

- Built-in support for server-side rendering (SSR) and static site generation (SSG).
- Simplifies API creation with serverless functions.
- Optimized for SEO and performance.
- Offers a powerful file-based routing system.

6. Nuxt.js

- Supports SSR, SSG, and client-side rendering out of the box.
- Provides automatic routing and file-based structure.
- Optimized for SEO and performance.
- Extensive module ecosystem for integrating features like analytics and authentication.

7. Ember.js

- Convention over configuration approach for rapid development.

- Includes built-in tools for routing, data layer management, and templating.
- Focuses on scalability and developer productivity.
- Provides an Ember CLI for managing dependencies and building projects.

C. DATA MANAGEMENT SYSTEM

Database Management Systems (DBMS) are software systems used to store, retrieve, and run queries on data. A DBMS serves as an interface between an end-user and a database, allowing users to create, read, update, and delete data in the database. They manage the data, the database engine, and the database schema, allowing for data to be manipulated or extracted by users and other programs. This helps provide data security, data integrity, concurrency, and uniform data administration procedures.

DBMS optimizes the organization of data by following a database schema design technique called normalization, which splits a large table into smaller tables when any of its attributes have redundancy in values. DBMS offer many benefits over traditional file systems, including flexibility and a more complex backup system (What is Database Management Systems (DBMS)? 2023). Database management systems can be classified based on a variety of criteria such as the data model, the database distribution, or user numbers. The most widely used types of DBMS software are relational, distributed, hierarchical, object-oriented, and network.

A distributed DBMS is a set of logically interrelated databases distributed over a network that is managed by a centralized database application. This type of DBMS synchronizes data periodically and ensures that any change to data is universally updated in the database (What is Database Management Systems (DBMS)?, 2023).

Hierarchical databases organize model data in a tree-like structure. Data storage is either a top-down or bottom-up format and is represented using a parent-child relationship. The network database model addresses the need for more complex relationships by allowing each child to have multiple parents. Entities are organized in a graph that can be accessed through several paths. Relational database management systems (RDBMS) are the most popular data model because of its user-friendly interface. It is based on normalizing data in the rows and columns of the tables. This is a viable option when you need a data storage system that is scalable, flexible, and able to manage lots of information.

Object-oriented models store data in objects instead of rows and columns. It is based on object-oriented programming (OOP) that allows objects to have members such as fields, properties, and methods.

There is a wide range of database software solutions, including both enterprise and open-source solutions, available for database management. Here are some of the most popular database management systems:

Oracle Database is a commercial relational database management system. It utilizes enterprise-scale database technology with a robust set of features right out of the box. It can be stored in the cloud or on-premises.

MySQL is a relational database management system that is commonly used with open-source content management systems and large platforms like Facebook, Twitter, and YouTube. Developed by Microsoft, SQL Server is a relational database management system built on top of structured query language (SQL), a standardized programming language that allows database administrators to manage databases and query data.

SQLite3 is a database engine written in the C programming language. It is not a standalone app; rather, it is a library that software developers embed in their apps. As such, it belongs to the family of embedded databases. It is the most widely deployed database engine, as it is used by several of the top web browsers, operating systems, mobile phones, and other embedded systems. Many programming languages have bindings to the SQLite library. It generally follows Postgre SQL syntax, but does not enforce type checking by default. This means one can, for example, insert a string into a column defined as an integer.

D. Mobile platforms

Android

Android is a mobile operating system that has been around for nearly 15 years. You'll primarily find it as the base operating system of phones and tablets worldwide. Additionally, other operating systems natively support Android applications, including Chrome OS and Windows 11. Search giant Google owns this mobile OS (Mobile platforms, frameworks & environments, 2023). However, the system is

open source, making it freely accessible to anyone, even for commercial use. This makes it very different from Apple's iOS, MACOS, and Microsoft's Windows, all closed-source platforms.

IOS

Android is by far the world's most popular operating system. Global Stats' Stat counter puts Android in the lead in mobile OS market share with a 69.74% as of January 2022. This dwarfs Apple's iOS by a significant margin, which is the second-most-popular mobile operating system globally at 29.49% (Mobile platforms, frameworks & environments, 2023). There are over 2.6million applications available from the official Google Play Store, but you can also sideload apps from the web. This variety makes these phones very powerful and customizable — but also susceptible to viruses and other types of malwares. IOS, an operating system from Apple, was originally developed for the iPhone. Later it was extended to support iPod Touch, iPad and Apple TV. Apple's App Store contains more than 500,000 applications and boasts more than 25 billion downloads collectively. It holds the reputation of intelligent UI creator which is based on the concept of direct manipulation, using multi-touch gestures.

2.2 CONCEPTUAL REVIEW

Here we are going to show how the different components explained above can interact with each other and produce an overview of our project.

In a typical web application architecture, the system comprises several layers that work together to deliver a seamless user experience. The following is a textual representation of a web application architecture:

- **User Interface (UI):**

The front-end layer where users interact with the application. Built with HTML, CSS, and JavaScript. Responsive design for optimal user experience across devices.

- **Presentation Layer:**

Handles the logic related to the presentation and user interface.

Includes front-end frameworks like React, Angular, or Vue.js.

Responsible for rendering data received from the server.

- **Application Logic Layer:**

Manages the core functionality of the application.

Utilises back-end frameworks like Node.js, Django, or Ruby on Rails.

Processes requests from the presentation layer and communicates with the database.

- **Web Server:**

Acts as an intermediary between the front-end and back-end.

Responds to HTTP requests from the client and routes them to the appropriate components.

- **API (Application Programming Interface):**

Defines how different software components should interact.

RESTful or GraphQL APIs are common for communication between the front-end and back-end.

Business Logic Layer:

- Implements the business rules and logic of the application.
- Validates and processes data before storing it in the database.

Database:

- Stores and manages the application's data.
- Relational databases like MySQL or PostgreSQL, or NoSQL databases like MongoDB are commonly used.

Data Storage:

- Includes file storage systems or cloud storage for storing user uploads, media files, etc.
- Authentication and Authorization:
- Manages user identity and access control.

- Ensures secure user authentication and authorization to access specific resources.

Security Layer:

- Implements security measures to protect against common web application vulnerabilities.
- Includes encryption, input validation, and other security protocols.

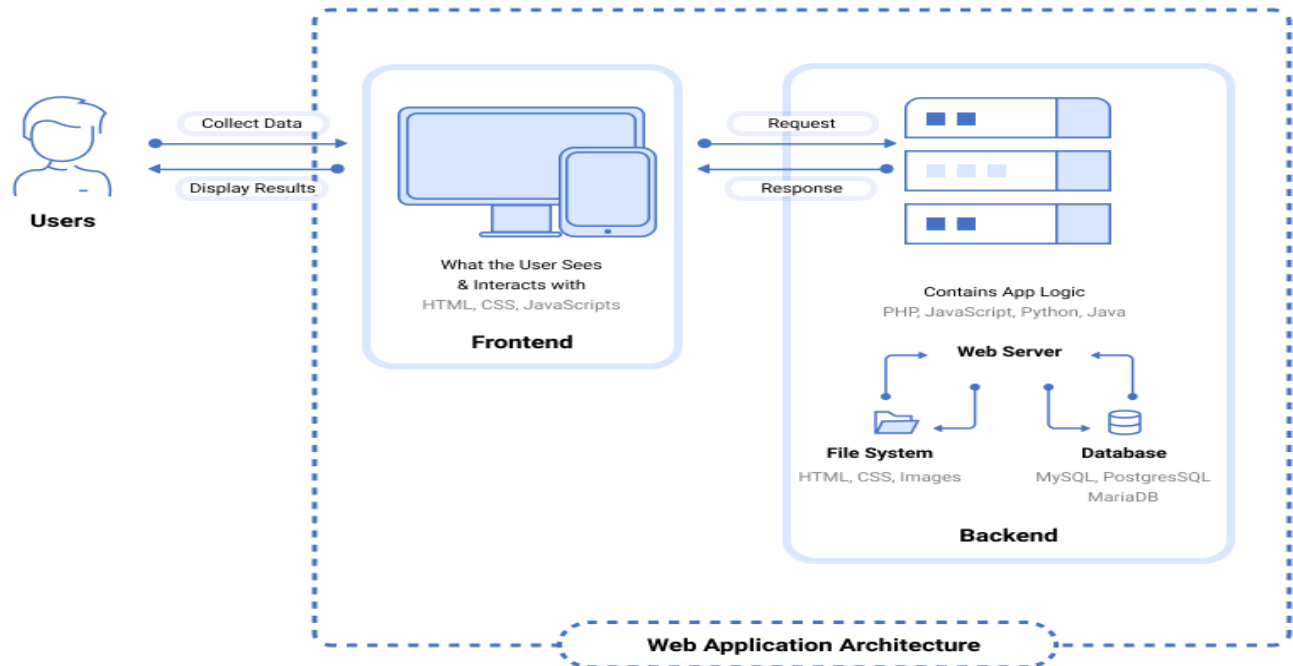


Figure 1 :3-tier architecture

source(Altwater, 2017)

CHAPTER THREE:

RESEARCH METHOD AND INTERNSHIP ACTIVITIES

INTRODUCTION

In order to guarantee the success and effectiveness of our project, it is necessary at this stage of our work to define precisely the methods adopted to develop the solution. This includes the modeling and enumeration of the various services that our system is supposed to offer to various users and the study of the various materials to be used.

3.1 CHOSEN ENGINEERING METHOD

When developing a software, we may choose any one of the following software engineering processes: waterfall model, V-Shape model, prototyping model, RAD, incremental, iterative, spiral, Agile. The choice of a model generally depends on the following factors: requirements, the date lines, risk factors,

availability of technical resources, and financial resources: we choose the prototyping model to develop our system: in this section we first describe the prototyping model before justifying our choice;

3.1.1 Prototyping model

The Prototype Model is a Software Development approach useful for projects with vague or changing requirements. It involves creating a preliminary version of the software product, called a prototype, that shows the basic features and functionality that the customers want.

The prototype is then tested and improved by the customers and the developers until the customers are happy with the final prototype, which forms the basis for the final product. The Prototype Model has some benefits, such as increasing customer involvement, reducing risks, and improving quality. However, it also has some drawbacks, such as being time-consuming, costly, and hard to document (The Knowledge Academy, n.d.-a).

A. TYPES OF PROTOTYPING MODEL

Prototyping Models are Software Development approaches that involve creating preliminary versions of the software product, called prototypes, that demonstrate the basic features and functionality that the customers want. There are four types of Prototyping Models:

1) Rapid Throwaway Prototyping

This technique offers a valuable method of exploring ideas and getting customer feedback for each of them. In this method, a developed prototype does not necessarily need to be a part of the ultimately accepted prototype. Customer feedback helps prevent unnecessary design faults, so the final prototype developed is of better quality.

The Rapid Throwaway Prototyping technique suits projects with unclear or changing requirements or when the developers want to experiment with different alternatives. The main advantage of this technique is that it allows the developers to identify and eliminate the flaws in the design quickly. The main disadvantage is that it can be wasteful and time-consuming, as the prototype is discarded after each iteration.

2) Evolutionary Prototyping

In this method, the prototype developed initially is incrementally refined based on customer feedback till it finally gets accepted. Compared to Rapid Throwing Prototyping, Evolutionary Prototyping offers a better approach that saves time and effort. This is because developing a prototype from scratch for every iteration of the process can sometimes frustrate developers.

The Evolutionary Prototyping technique is suitable for projects with stable and well-defined requirements or when the developers want to deliver a working product as soon as possible. The main advantage of this technique is that it allows the developers to improve the quality and functionality of the product over time. The main disadvantage is that it can be challenging to document and maintain as the prototype undergoes

3) Incremental Prototyping

In this type of Incremental Prototyping, the final expected product is broken into different small pieces of prototypes and developed individually. Ultimately, when all individual pieces are appropriately developed, the various prototypes are collectively merged into a single final product in their predefined order. It's a very efficient approach that reduces the complexity of the development process, where the goal is divided into sub-parts, and each sub-part is developed individually.

The time interval between the projects's beginning and final delivery is substantially reduced because all system parts are prototyped and tested simultaneously. Of course, there might be the possibility that the pieces just do not fit together due to some lack of ness in the development phase this can only be fixed by careful and complete plotting of the entire system before prototyping starts.

The Incremental technique is suitable for projects with modular and independent components or when the developers want to deliver a functional product in stages. The main advantage of this technique is that it allows the developers to focus on one part of the system at a time and get customer feedback. The main disadvantage is that it requires careful and complete planning of the entire system before prototyping starts, as any changes in the requirements or design can affect the integration of the prototypes.

4) Extreme Prototyping

This method is mainly used for Web Development. It consists of three sequential independent phases:

a) A basic prototype with all the existing static pages is presented in HTML format in this phase. This phase is used to create the user interface and navigation of the web application and get feedback from the customers. It allows the developers to design and test the layout and appearance of the web pages quickly. However, a basic prototype does not include any functionality or data processing of the web application.

b) In the second phase, functional screens are made with a simulated data process using a prototype services layer. This phase is used to create the web application's functionality and logic and get customer feedback. The phase allows the developers to quickly implement and test the features and behavior of the web application, although without any actual data or services of the web application. (The Knowledge Academy, n.d.-a)

c) This is the final step, where all the services are implemented and associated with the final prototype. This phase is used to create the data and services of the web application and integrate them with the user interface and functionality.

This Extreme Prototyping method makes the project cycling and delivery robust and fast. Moreover, it keeps the entire developer team focused and centralized on product deliveries rather than discovering all possible needs and specifications and adding necessitated features. This technique is suitable for projects with web-based and dynamic applications or when the developers want to use an agile and iterative approach to Web Development. The main advantage of this technique is that it allows the developers to rapidly prototype and test the web application and get customer feedback. The main disadvantage is that implementing and integrating all the services with the final prototype can be complex and risky.

B. PHASES OF THE PROTOTYPING MODEL

The Prototype Model is characterized by its iterative nature, which allows for the development of a functional prototype followed by continuous refinement based on feedback.

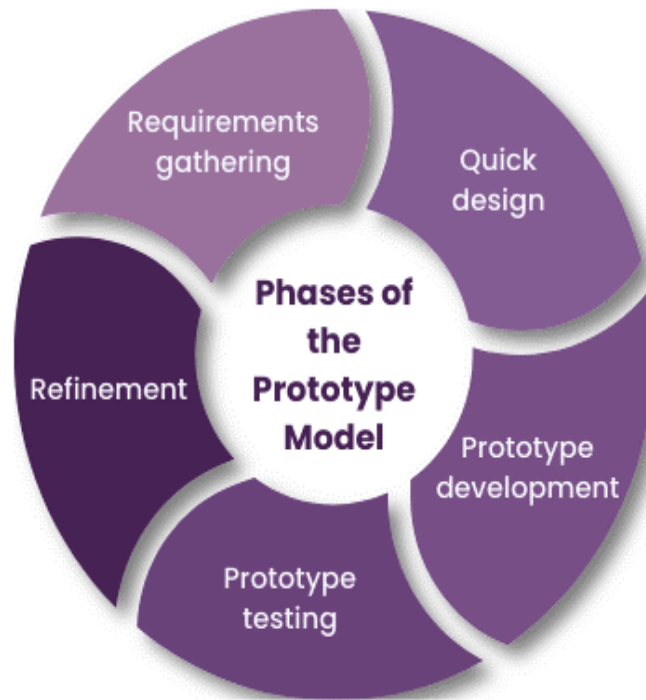


Figure 2 *GLPI admin dashboard functionalities*

Source (The Knoledge Academy, a) n.d.-)

- 1. Analysis and Requirements gathering:** During this phase, developers engage in in-depth discussions with stakeholders to identify the primary objectives of the software. However, unlike traditional methods, where all requirements are gathered up front, the Prototype Model focuses on capturing the essential features and functionalities that will be incorporated into the initial prototype.
- 2) Quick design:** With the initial requirements in hand, developers create a quick and simplified design of the software's user interface and core functionalities. This design serves as a blueprint for the prototype, outlining the basic layout, user interactions, and essential features. The goal is to establish a foundation that can be built upon in subsequent phases.
- 3) Prototype development:** Based on the quick design, developers begin creating the actual prototype of the software. This prototype is a working model that focuses on demonstrating the core features

and functionalities of the software. It might need some of the intricate details and advanced features that would be present in the final product.

4) Prototype testing: Once the prototype is developed, it undergoes testing to identify any issues, bugs, or areas of improvement. This testing phase involves both developers and clients working collaboratively to evaluate the prototype's functionality and user experience. Clients interact with the prototype, providing real-time feedback on its usability and alignment with their requirements.

5) Refinement: Feedback obtained from prototype testing forms the basis for the refinement phase. Developers analyze the feedback and begin making necessary changes and improvements to the prototype.

These changes involve addressing identified issues, adding missing functionalities, and enhancing the user interface. The prototype is iteratively (The Knowledge Academy, n.d.-a) refined based on client feedback, ensuring that it becomes a more accurate representation of the final product with each iteration.

C. ADVANTAGES OF THE PROTOTYPING MODEL

- The Prototype Model in Software Engineering offers several distinct advantages. It promotes enhanced collaboration between developers and clients, promoting active involvement and feedback throughout development.
- Early detection of issues is facilitated by creating a functional prototype, enabling timely identification and resolution of potential problems. The model's inherent flexibility and iterative approach accommodate changing requirements and allow for continual improvements, ensuring that the final product aligns more accurately with client expectations.
- The Prototype Model facilitates a deeper understanding of project requirements as developers and clients gain more precise insights into the software's functionalities and design, resulting in a more refined and effective product.

D. DISADVANTAGES

- The flexibility that allows for iterative improvements can also lead to scope creep, where evolving client demands may extend project timelines and increase complexity. Incomplete initial requirements might hinder the effectiveness of the prototype, requiring additional iterations to address client needs fully.
- There's a risk of unrealistic expectations, as clients might need to understand the prototype as the final product, potentially leading to disappointment if outcomes differ from prototype functionality or design.

- The iterative nature of the model can sometimes lead to time and resource constraints, as multiple refinement cycles may consume more resources than initially planned.

3.1.2 Justification of choice

- Digital software prototyping is more than just an average 3D model of your product. It mimics operation, functionality and allows testing of the final product before physically constructing it. By building a working simulation of the actual application, designers can effectively visualize, optimize and iterate the product digitally, saving time and money.
- Procrastination can badly affect businesses, especially in complex work environments where one has to meet tight deadlines. Digital prototyping enables quick action and quality decision making. Designers can start immediately, choosing from a number of prototyping tool available. Available, making changes and incorporating new requirements and ideas with ease.
- Digital prototyping enables users to continuously amend an agile model to reach the intended product design. It allows changes and refinements in quick succession, throughout the process. Using this methodology, designers can plan, optimize, validate, and visualize products beforehand without physically developing the actual product. This allows early identification and correction of faults in the product design. This technology also allows testing the model for robust qualities by subjecting it to load conditions, as a means to assess the product's behavior and interaction.
- Cost savings is one of the main driving forces behind incorporating digital software prototyping. Prototyping helps in minimalizing actual production time, thereby shortening the development cycle. It enables concurrent crafting among team members too, reducing the resources and labor required. Prior to digital prototyping, software engineers had to reach the end of the development phase before testing their designs. A single glitch could in extremis lead to the entire cycle being repeated incurring exaggerated and unnecessary costs. Today, design engineers can deploy digital software prototyping skills and recreate functional models, without incurring significant additional expenditure.

3.1.3 APPLICATION OF THE PROTOTYPING MODEL

Though we choose the prototyping model but we specifically choosed the rapid Throwaway Prototyping model.

1. Analysis and Requirement gathering

To design this system, it was essential for us to collect the necessary information from specialists in the field. After structuring the information collected, we noticed that we needed the following things:

- A database for IT equipment's
- Recording of the different maintenance tasks
- Inserting new IT equipment's into the database
- Tracking of IT equipment's
- Monitoring the activity of the IT technicians.
- Statistic view of the different equipment and status.

Before talking about actual operation of the system, it is necessary to first define the functionalities that will be implemented within the system. So, this step will describe what we expect from our application. Then all of this will be modeled as a diagram using UML modeling language.

Management structure

- ❖ **Description:** An administrator records the maintenance activity and records IT equipment in the system.
- ❖ **Modules**
 - **Inventory management**
 - **IT technician task management**

Maintenance request management

- **Description** : A user will store information's on he's request for maintenance
- **Sub-modules**
 - **Management of maintenance tasks**
 - **Equipment inventory**

Maintenance management

After identifying the requirements, we came out with some UML diagrams to model the requirements

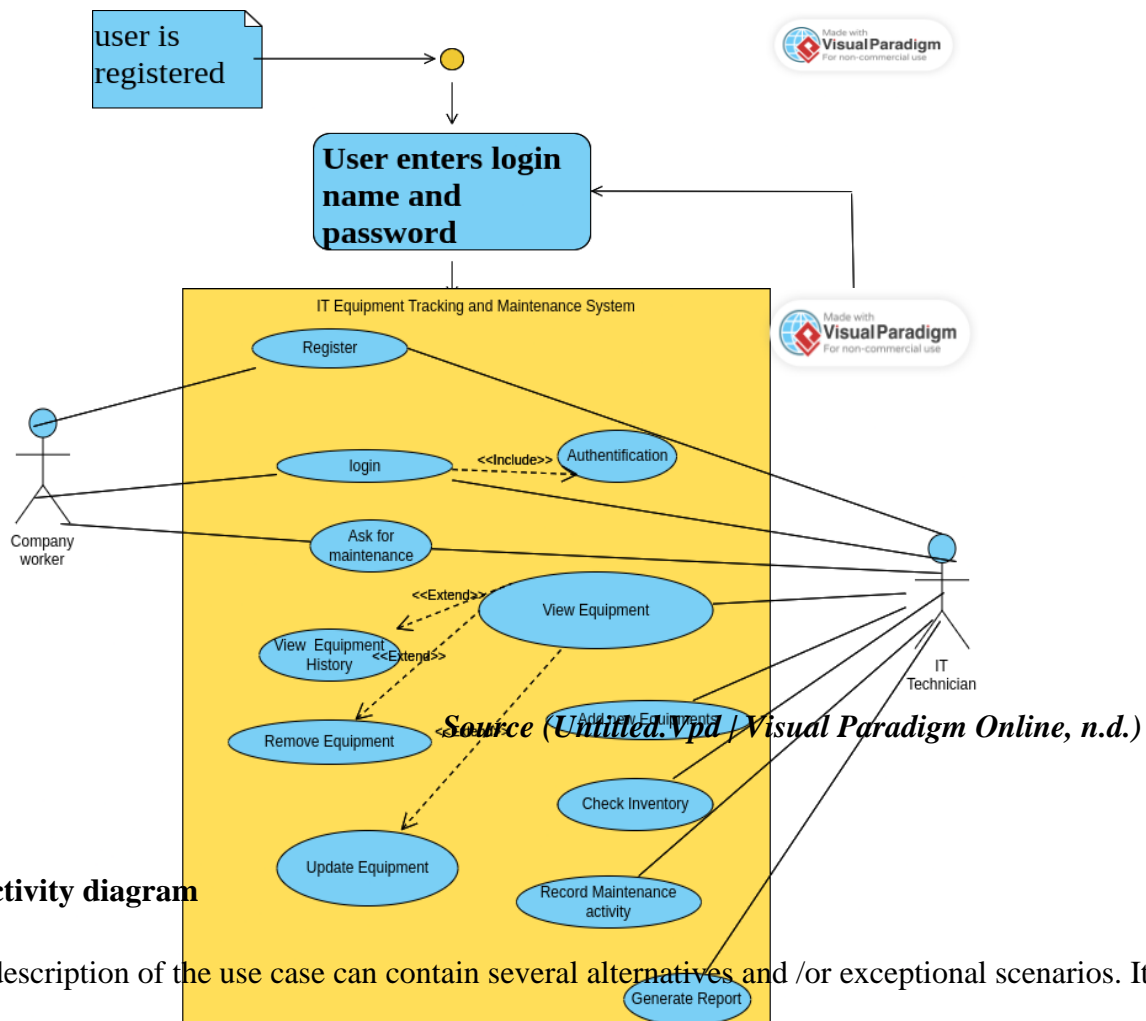
2. Quick design

Here we came out with the different functionalities for every actors in our system, not the actors have the same roles and activities the system. We came out with our use case diagram that showed all the different functionalities our system in it and we came out with a basic user interface that served us for our future product. All these designs are represented in the figure and figure

a) Use case diagrams

DESIGN AND IMPLEMENTATION OF A BAKERY MANAGEMENT SYSTEM

Use case diagram is a representation of a user's and administrator's interaction with the system that shows the Relationship between the user, administrators and the different use cases in which the user is involved.



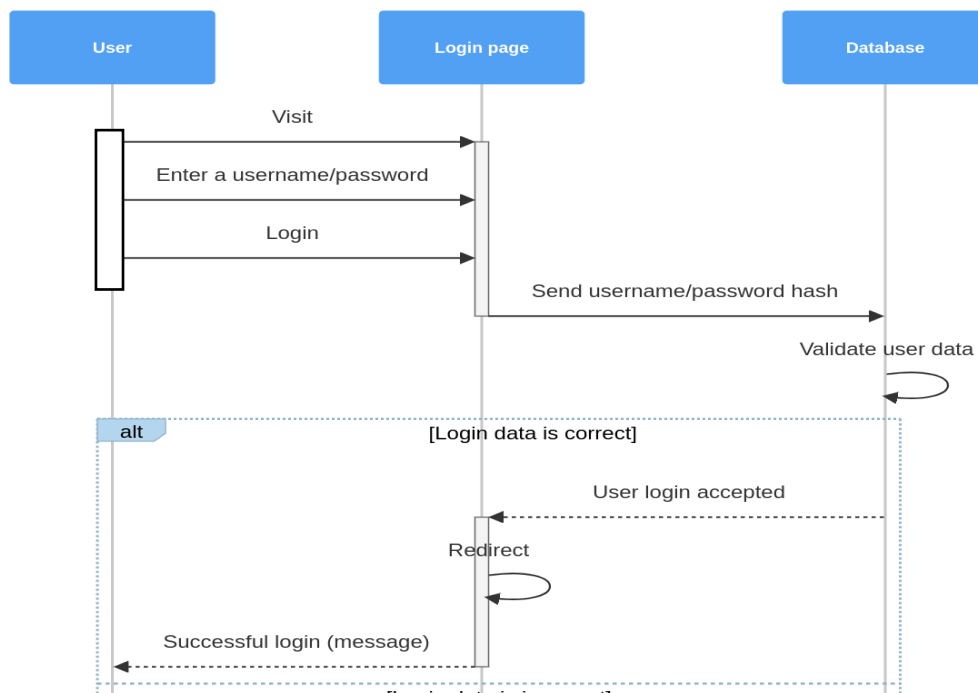
b) Activity diagram

The description of the use case can contain several alternatives and /or exceptional scenarios. It is Therefore difficult to have a vision of all the actions. The activity diagram is a graphical means to Give these overviews.

Figure 4 Activity diagram Source (Untitled.Vpd / Visual Paradigm Online, n.d.)

c) Sequence diagram

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.



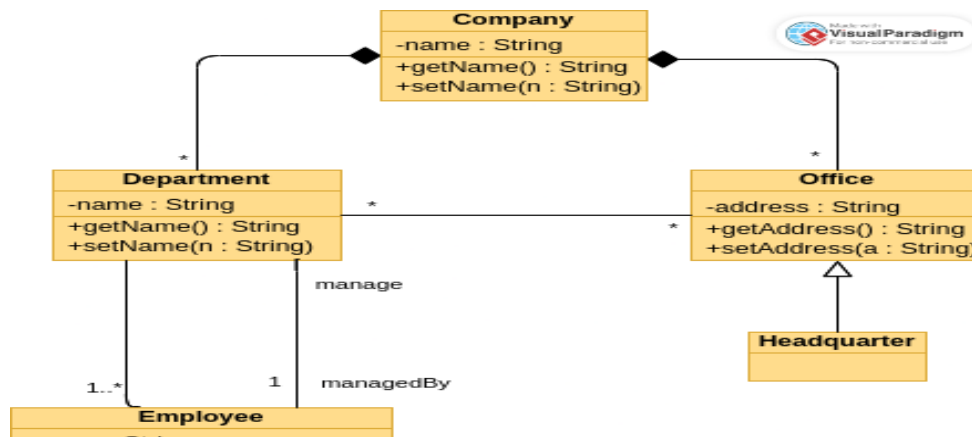


Figure 5 Sequence diagram Source (Untitled.Vpd / Visual Paradigm Online, n.d.)

The figure above represents a sequence diagram concerning the use case authentication. It permits all interaction existing between the actors to permit an authentication.

d) Class diagram: A class diagram shows the internal structure of the system. It permits to deliver an abstract representation of the object of the system which will interact together to realize a use case. It is a static view because we don't take into account the time factor in the behavior of the system. The class diagram models concept of the fields of the application in the context of implementation of an application.

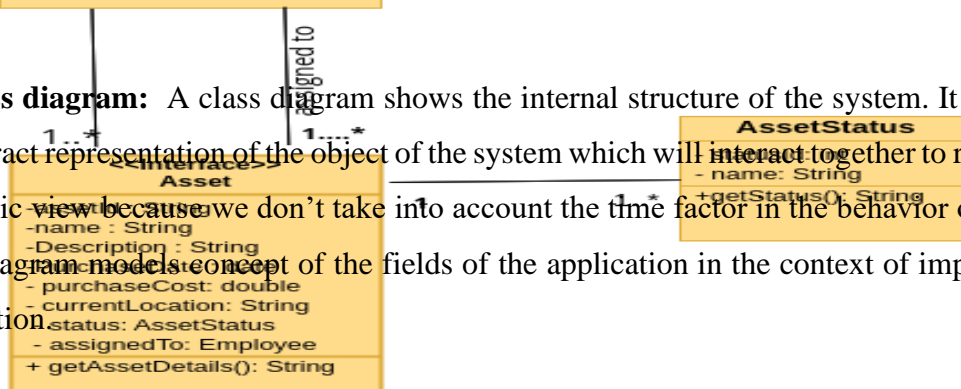


Figure 6 : Class diagram Source (Untitled.Vpd / Visual Paradigm Online, n.d.)

C. Prototype development

Here we started implementation the back-end of our features of our graphical user interface that we designed for it to operate its basic operations. We made use of algorithms to ease our work. This is the phase where most of the coding took part. This is where most of the coding and development took place.

Development technologies

Material (hardware) used for development

To carry out our work, we had at our disposal:

- A HP 250 G8 notebook laptop
- A Big Linux operating system (64 bit)
- A 15-1035G1 CPU @ 3.6 GHz processor
- 8.00 GB RAM memory
- A 1 TB 540 RPM SATA hard drive
- Software used for development
- vs.code IDE(version 1.48)

Client side

1) Framework

- Django 5.0
- Bootstrap: JavaScript framework to facilitate the design of our web pages and validation
- J query: JavaScript framework to simplify the coding of our client-side script

2) Languages

- Python 3 : to create classes and modules
- HTML 5: for formatting web pages
- CSS 3: for the style of our web components
- JavaScript: for client-side interaction

3) Database server

On the web server side

- Python 3: it is a server scripting language
- SQLite3: As our database

D. Prototype testing

Testing which is the process of evaluating and verifying that a software product or application does what it is supposed to do. And for this case it was tested through browsers by some Workers in the domain.

E. Refinement

After testing, the feedbacks came and we had to change things like the login page, the different privileges to the different users.

3.2 INTERNSHIP ACTIVITIES

During this period of time, we did many activities during the internship even if some of them were not oriented towards our field of academic study but though we still gain enough knowledge and experience which include:

3.2.1 Activities carried out in the different departments

A. Installing, configuring and testing new hardware components such as motherboards, processors, memory, hard drives, etc.:

Installing, configuring, and testing new hardware components such as motherboards, processors, memory, hard drives, etc. involves several steps. Here Is a General outline of the process :

- 1. Preparation:** Ensure you have the necessary tools and equipment for the installation, such as screwdrivers, thermal paste, and anti-static wristbands. It's important to handle hardware components with care to avoid damage from static electricity.
- 2. Research:** Familiarize yourself with the specific hardware components you are installing. Read the manufacturer's documentation, user manuals, and online guides to understand the installation process and any specific requirements.
- 3. Safety:** Before working on any hardware components, make sure to power off and unplug your computer from the electrical outlet. This Will prevent any accidental damage or electrical shock.
- 4. Installation:** Follow the manufacturer's instructions to install each hardware component correctly. This may involve removing the computer case, connecting cables, inserting components into slots or sockets, and securing them in place with screws or clips.
- 5. Configuration:** Once the hardware components are physically installed, you may need to configure them through software settings or BIOS (Basic Input/Output System) options. This can include adjusting settings for memory timings, storage devices, or enabling/disabling certain features.
- 6. Testing:** After installation and configuration, it's important to test the new hardware components to ensure they are functioning correctly. This can involve running diagnostic tools, stress tests, or benchmarking software to verify performance and stability.
- 7. Troubleshooting:** If you encounter any issues during installation or testing, refer to the manufacturer's documentation or online resources for troubleshooting steps. It's also helpful to seek assistance from knowledgeable individuals or online communities.

Remember that working with hardware components requires caution and attention to detail. If you are unsure about any step of the process or lack experience in handling computer hardware, it's recommended to seek professional assistance.

B. Diagnosing and repairing hardware problems such as faulty power supplies, fans, cables, etc.

Diagnosing and repairing hardware problems such as faulty power supplies, fans, cables, etc. can be a challenging task. Here are some General steps to follow :

- 1. Identify the problem:** The first step is to identify the problem with the hardware component. This can involve running diagnostic tests, checking error messages, or observing unusual behavior.
- 2. Research:** Once you have identified the problem, research the specific hardware component and its symptoms. This can involve reading manufacturer documentation, online forums, or consulting with experts.
- 3. Safety:** Before working on any hardware components, make sure to power off and unplug your computer from the electrical outlet. This will prevent any accidental damage or electrical shock.
- 6. Prevention:** To prevent future hardware problems, make sure to keep your computer clean and free of dust, use surge protectors to protect against power surges, and perform regular maintenance such as updating drivers and firmware.

Remember that working with hardware components requires caution and attention to detail. If you are unsure about any step of the process or lack experience in handling computer hardware, it's recommended to seek professional assistance.

C. Updating and upgrading hardware drivers and firmware

To update and upgrade hardware drivers and firmware, you can follow these steps (Windows update tips, 2024):

D. Performing preventive maintenance and cleaning of hardware devices

Performing preventive maintenance and cleaning of hardware devices is crucial for ensuring their longevity and optimal performance, we performed this preventive maintenance in all HEVECAM.SA departments and its resources. Here are some general steps you can follow:

- 1. Regular Inspection:** Conduct regular inspections of your hardware devices to identify any signs of wear, damage, or accumulation of dust and debris.
- 2. Cleaning:** Use appropriate cleaning tools such as compressed air, lint-free cloths, and cleaning solutions to remove dust, dirt, and grime from the surfaces of your hardware devices. Be sure to power off and unplug the devices before cleaning them. We also did software cleaning by deleting all the temporary and used files in the computer so as to make the computer robust and more efficient at work.
- 3. Cable Management:** Organize and secure cables to prevent tangling, reduce strain on connectors, and improve airflow within the device.

E. Detecting and repairing hardware issues exposer

Detecting and repairing hardware issues can be a complex task, but there are several steps you can follow to troubleshoot common problems. Here are some General guide lines:

- 1. Check Device Manager:** Device Manager is a built-in Windows tool that can help you identify hardware issues. You can access it by searching for "Device Manager" in the Start menu. Look for any devices with a yellow exclamation mark or red "X" next to them, as these indicate potential problems.
- 2. Update Drivers:** Outdated or incompatible drivers can cause hardware issues. Visit the manufacturer's website for your hardware components and download the latest drivers. You can also use Windows Update to check for driver updates.
- 3. Run Hardware Troubleshooters:** Windows includes built-in troubleshooters that can automatically detect and fix common hardware problems. To access them, go to Settings> Update & Security > Troubleshoot.

4. Test Hardware Components: Use diagnostic tools to test individual hardware components such as memory, hard drives, and graphics cards. Many manufacturers provide diagnostic software that can help identify specific issues.

5. Consult Manufacturer Documentation: If you are experiencing issues with a specific hardware component, consult the manufacturer's documentation or support resources for troubleshooting steps specific to that device.

Remember, hardware issues can be complex, and it may be necessary to seek professional assistance if you are unable to resolve the problem on your own

F. Configuration and installation of security cameras

Installing and configuring security cameras can be a complex process, but it can be broken down into several steps. Here are some general guidelines:

1. Research: Familiarize yourself with the specific security cameras you are installing. Read the manufacturer's documentation, user manuals, and online guides to understand the installation process and any specific requirements.

2. Planning: Determine the number of cameras you need, their location, and the type of camera that best suits your needs. Consider factors such as lighting conditions, field of view, and resolution.

3. Installation: Follow the manufacturer's instructions to install each camera correctly. This may involve mounting the camera, connecting cables, and configuring settings.

4. Testing: After installation and configuration, it's important to test the cameras to ensure they are functioning correctly. This can involve running diagnostic tools or monitoring footage to verify performance and stability.

5. Maintenance: Regularly clean and inspect your cameras to ensure they are free from dust and debris. Check for any signs of wear or damage and replace any faulty components as needed.

6. Security: Ensure that your cameras are secure by using strong passwords, enabling encryption, and keeping firmware up to date.

Remember that working with security cameras requires caution and attention to detail. If you are unsure about any step of the process or lack experience in handling security cameras, it's recommended to seek professional assistance. To configure a camera, you need to follow the following steps.

Step 1: connect the camera to a power supply

Step 2: connect your camera and your computer via an ethernet cable

Step3: open your browser and enter the default Ip 192.168.1.108 to enter in the login interface of the camera

Step 4: change the login password and administrator name for your personal use, also change parameters as the camera Ip and port so it should be in your network. See images below.

G. Stock inventory

Inventory is the accounting of items, component parts, and raw materials that a company either uses in production or sells. It is an essential aspect of managing a company's assets and operations. Stock inventory, also known as stock control, refers to the process of tracking and managing the quantity, location, and value of inventory items. It involves keeping accurate records of stock levels, monitoring stock movements, and ensuring that the right amount of stock is available when needed.

H. Documenting and reporting hardware issues and solutions

Documenting and reporting hardware issues and solutions is an important aspect of maintaining a well-functioning IT infrastructure. Here are some suggestions on how to approach this task:

1. Hardware Issue Report Form: Consider using a standardized hardware issue report form to gather information about the reported issues. This can help ensure that all necessary details are captured consistently. You can find free templates online that you can customize to suit your specific needs.

2. Technical Documentation: Create technical documentation that outlines common hardware issues, their symptoms, and potential solutions. This documentation can serve as a reference for IT staff when

troubleshooting problems. It's important to keep this documentation up to date as new issues arise and new solutions are discovered.

3. Error Code Analysis: Familiarize yourself with error codes generated by hardware devices and use them as a starting point for troubleshooting. Error codes can provide valuable information about the nature of the problem and potential solutions.

I. Fusion of Optical fibers

Fusion splicing is the process of joining two optical fibers end-to-end using heat to create a single long fiber. The goal is to fuse the two fibers together in such a way that light passing through the fibers is not scattered or reflected back by the splice, and so that the splice and the region surrounding it are almost as strong as the intact fiber. This is achieved by melting the ends of the fibers and fusing them together, resulting in a permanent joint known as a fusion splice. The heat source used for fusion splicing is typically an electric arc, but other methods such as lasers, gas flames, or tungsten filaments can also be used.

Fusion splicing is widely used in the telecommunications industry for creating reliable and low-loss connections between optical fibers. It ensures efficient transmission of light signals without significant signal degradation or loss. Fusion splicing is commonly performed using specialized equipment called fusion splicers, which provide precise control over the fusion process.

J. Design and development of a weighbridge system

Weigh bridge or railroad scale is a large set of scales, usually mounted permanently on a concrete foundation that is used to weigh entire rail or road vehicles and their contents. By weighing the vehicle both empty and when loaded, the load carried by the vehicle can be calculated.

3.2.2 BENEFITS AND ACHIEVEMENTS DERIVED FROM THE INTERNSHIP

Some of the achievements I accomplished during my internship were:

- Successfully installing and configuring over 100 new hardware components for various HEVECAM.SA workers.
- Resolving over 80% of hardware, software and network problems within the first hour of diagnosis

- Receiving positive feedback and appreciation from workers and supervisors for my work quality and professionalism.
- Learning new skills and techniques in computer hardware maintenance, networking and troubleshooting.
- Improving my communication, teamwork, problem-solving and worker service skills.
- Building a network of contacts and mentors in the field of computer hardware, networking and software solutions.
- Complete resolution of workers problems without the supervision or presence of my professional supervisor.

3.3 CHALLENGES DURING THE INTERNSHIP

Some of the challenges I faced during my internship were:

- a. Dealing with complex and unfamiliar hardware, software and Network issues that required extensive research and analysis, such as:**
 - Optical fibers rupture
 - Human resources department printer who is not functioning
 - The sudden shutdown of cameras at the industrial departments of HEVECAM.SA
 - Malfunctioning of printer drivers in some computers
 - Wrong configuration and set up a network in the network system of HEVECAM.SA
 - Very low band width of internet in the network
- b. Working under pressure and meeting tight deadlines for hardware repairs and installations.**
- c. Handling HEVECAM.SA complaints and dissatisfaction with hardware, software and network performance or quality.**
- d. Adapting to different work environments and cultures in different departments and services of HEVECAM.SA.**

CHAPTER FOUR:

PRESENTATION, ANALYSIS AND INTERPRETATION OF RESULTS.

4.1 Results

During the process of development We developed 3 prototype till date and that's the 3 prototype I'm going to present below.

4.1.2 First prototype

It was just a simple interface with no login page and was a desktop application with a local database made with SQLite3 and python3.

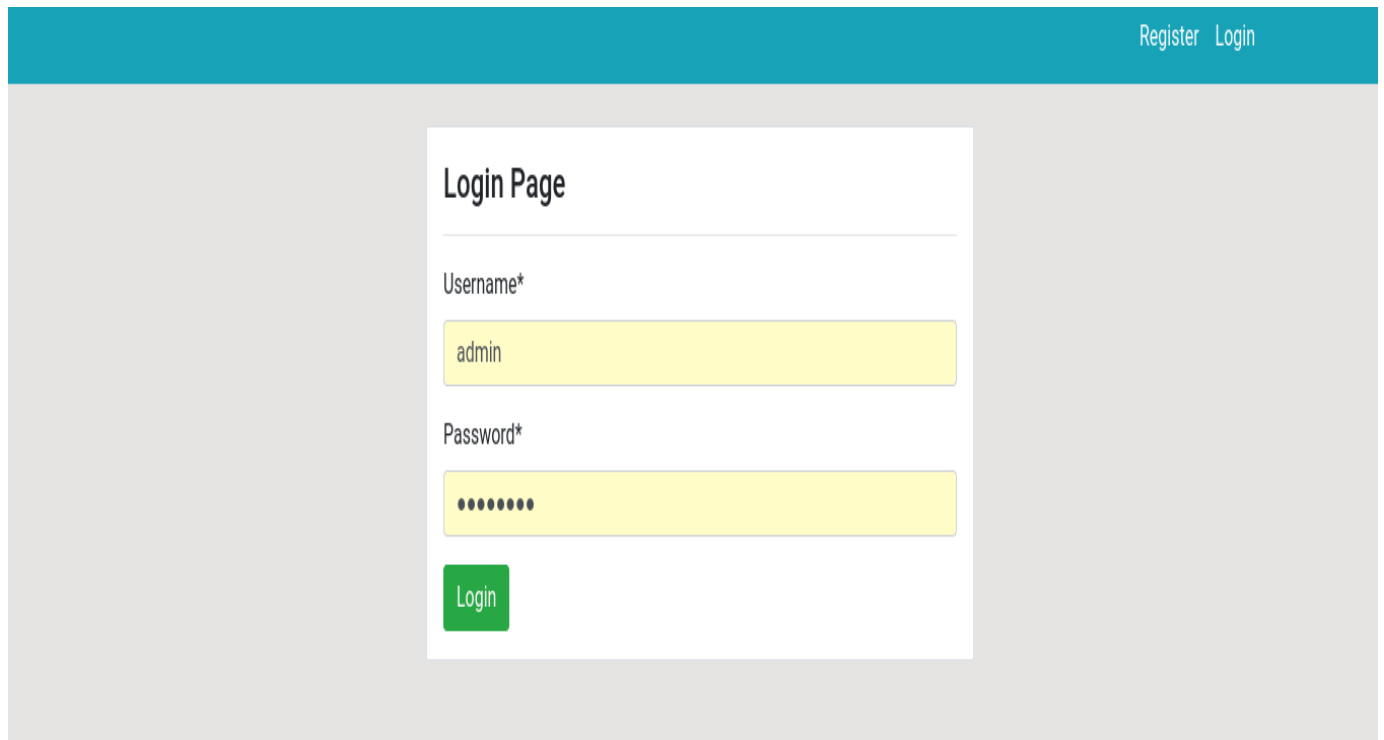
4.1.3 Second prototype

He was more advanced than the first one had a login page with different tabs and button to navigate through it and it was more user friendly than the first one though it was still a desktop application

The software had some functions like;

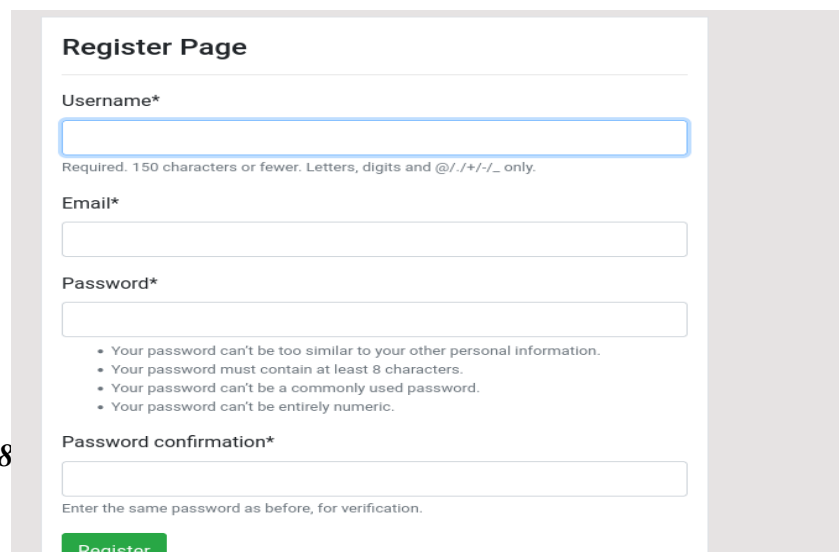
4.1.3 Prototype 3

1. Login page: Where the user enters or logs into the system



The screenshot shows a web application interface with a teal header bar containing the links "Register" and "Login". The main content area is light gray and features a white "Login Page" form. The form has a title "Login Page", a "Username*" field with the text "admin", a "Password*" field with masked characters, and a green "Login" button.

2. Registration Form



The screenshot shows a web application interface with a light gray background. The main content area features a white "Register Page" form. The form has a title "Register Page", a "Username*" field with a blue border and a note "Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.", an "Email*" field, a "Password*" field with a list of password requirements, and a "Password confirmation*" field with a note "Enter the same password as before, for verification." A green "Register" button is at the bottom.

Register Page

Username*

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Email*

Password*

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

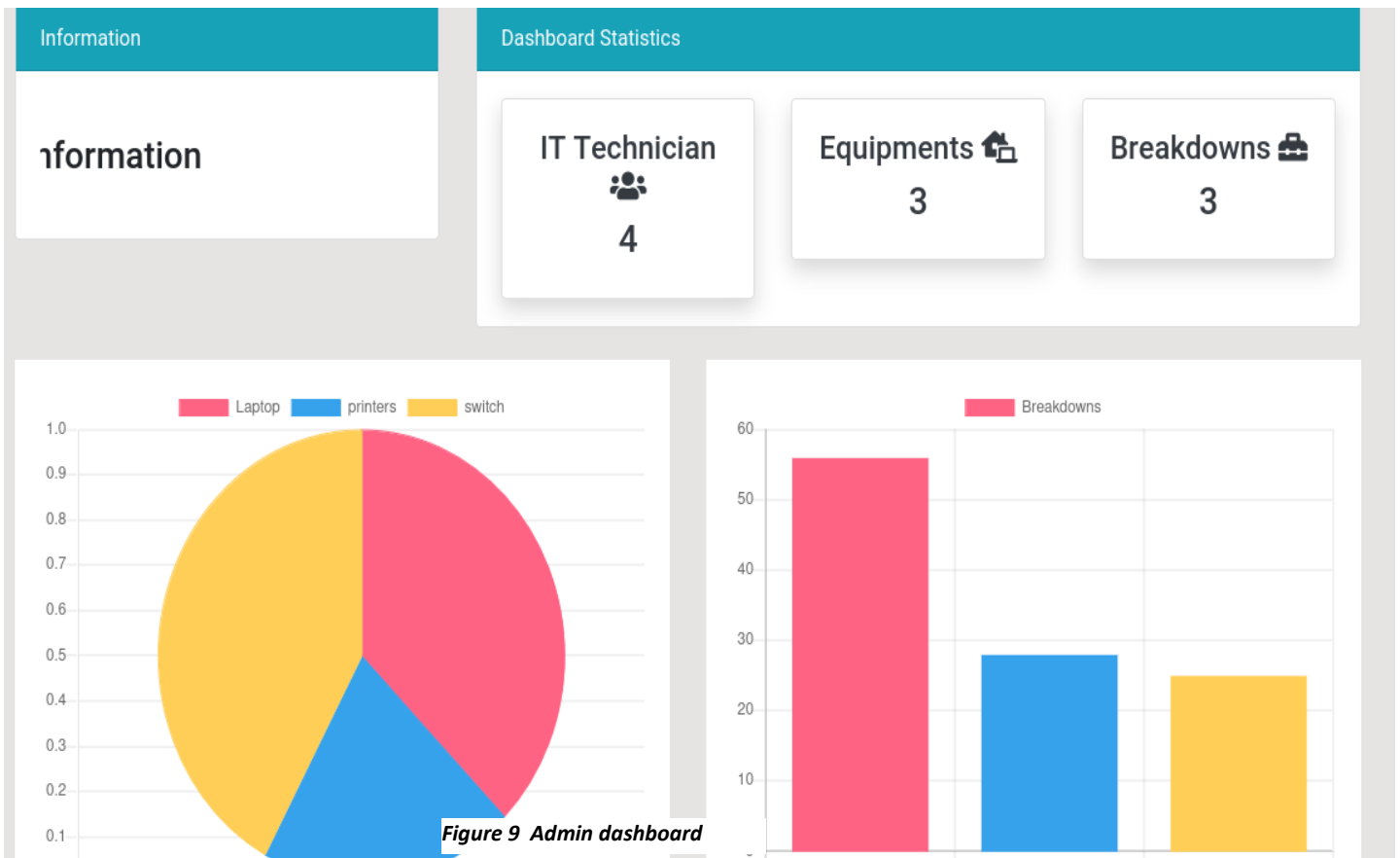
Password confirmation*

Enter the same password as before, for verification.

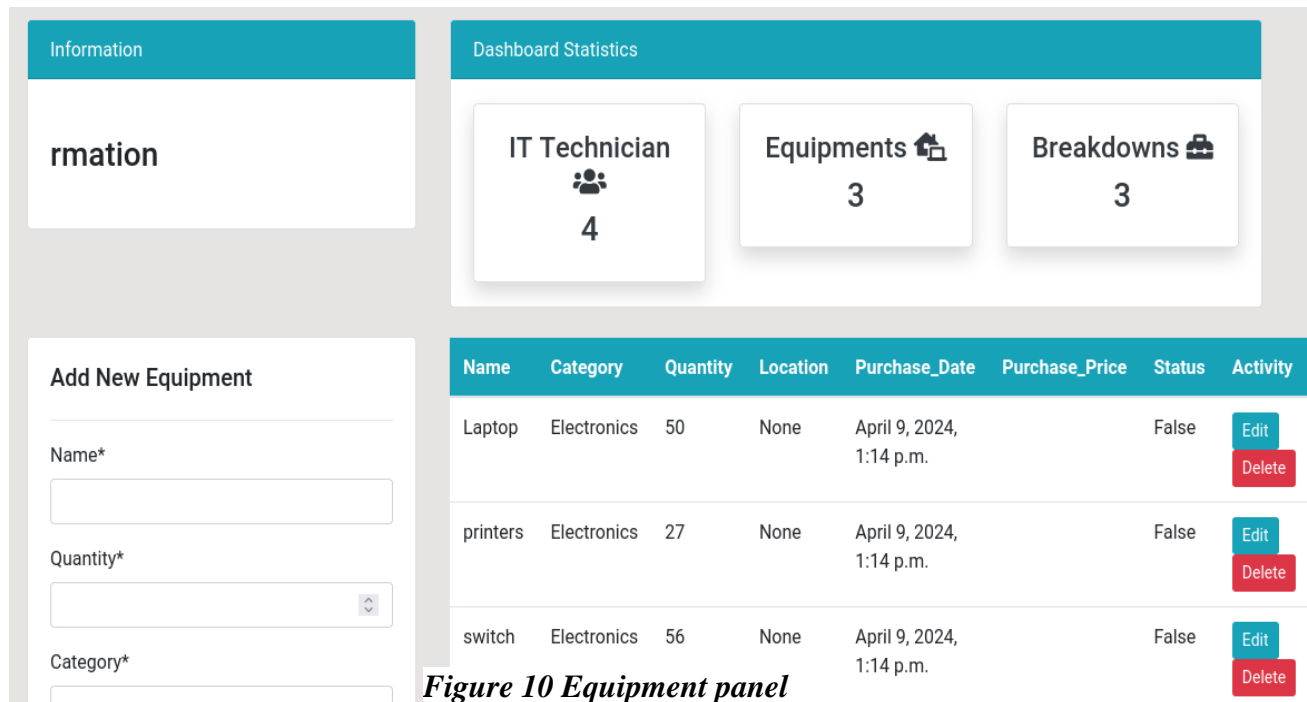
Register

Figure 8

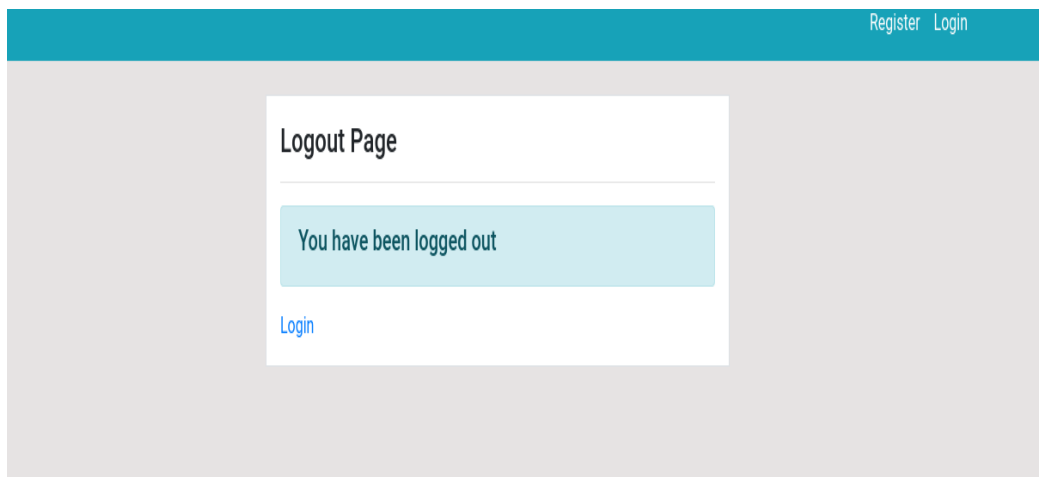
2. Admin dashboard



3. Equipment : where you can view , delete , add equipment



5. Log out



CHAPTER FIVE:**SUMMARY OF FINDINGS AND RECOMMENDATIONS****INTRODUCTION**

Our study focused on the creation of a web application to computerize the maintenance management process. To do this, we carried out a preliminary study on existing systems followed by the modeling of our system, modeling guided by the UML method through some of these diagrams, which allowed us to set up a system semi-functional for the hour, the progress of which is estimated at 93.5%. The first task we set up out to do was framing our project with the Project Director with whom we decided on certain major axes of the project such as the deployment environment, the technologies and languages to be used. We then had to agree to build the conceptual data model which is the core of the application and which would allow us to build our database from which the implementation would then really take on its full meaning.

5.1 Summary of findings

Our result revealed the following;

- There were many old devices in the company that were unused.
- Many workers who came for maintenance had mostly the problem of RAM (random access memory) failure.
- Most of the softwares installed on computers were bought rather than been developed by the software engineers of the company.
- Most workers had no little knowledge on recent technologies.
- The management of maintenance task was done in a book and recorded with a pen.

5.2 Recommendation

- **HEVECAM.SA** should invest more in training and development programs for IT staff to keep them updated with the latest trends and technologies in computer hardware and software solutions.
- The IT support department should implement a more efficient and standardized system for documenting and reporting hardware issues and solutions to improve its service quality and accountability.
- Future interns should take advantage of the learning opportunities and resources available at **HEVECAM.SA** to enhance their skills and knowledge in computer hardware maintenance and troubleshooting. They should also seek feedback and guidance from their supervisors and colleagues to improve their work performance.

5.3 Conclusion

In conclusion, my internship **HEVECAM.SA** was a valuable and rewarding experience that helped me achieve my objectives and prepare me for my future career. I gained practical skills and knowledge in computer hardware maintenance, software, networking and troubleshooting, applied the theoretical concepts and principles learned in my computer engineering program to real-world situations, developed professional and interpersonal skills such as communication, teamwork, problem-solving and worker service, explored the career opportunities and challenges in the field of computer hardware and software solutions. I would like to thank **HEVECAM.SA**, the information and technology support department and especially my supervisor **Mr. Bityeki Micheal** for giving me this opportunity and supporting me throughout my internship. I hope to continue working with them In the future.

REFERENCES

Online Newspaper Articles:

Anshul_A. (2024, March 01). Introduction to visual studio -GeeksForGeeks. Retrieved from GeeksForGeeks.org: <https://www.geeksforgeeks.org/introduction-to-visual-studio/>

Windows update tips. (2024, January). Retrieved from stackoverflow: www.stackoverflow.com

("GLPi Review 2024," n.d.-a; 6 Reasons Why You Must Deploy Software Prototyping, n.d.-a; 6 Reasons Why You Must Deploy Software Prototyping, n.d.-a; TheKnowledgeAcademy, n.d.-a; "GLPi Review 2024," n.d.-b; "GLPi Review 2024," n.d.-c; "GLPi Review 2024," n.d.-d; GLPi Reviews - Pros & Cons, Ratings & More, n.d., n.d.-b, n.d.-c, n.d.-d; Altvater, 2017; Untitled.Vpd | Visual Paradigm Online, n.d.)

6 Reasons Why You Must Deploy Software Prototyping. (n.d.). Retrieved March 25, 2024, from <https://www.rezaid.co.uk/post/6-reasons-why-you-must-deploy-software-prototyping>

Altvater, A. (2017, April 7). Top Load Testing Tools: The Best Tools for Load Testing Reviewed & Compared. Stackify. <https://stackify.com/top-load-testing-tools/>

GLPi Review 2024: Features, Pros, Cons and Pricing. (n.d.-a). Tech.Co. Retrieved March 23, 2024, from <https://tech.co/asset-tracking/glpi-review>

GLPi Review 2024: Features, Pros, Cons and Pricing. (n.d.-b). Tech.Co. Retrieved March 25, 2024, from <https://tech.co/asset-tracking/glpi-review>

GLPi Review 2024: Features, Pros, Cons and Pricing. (n.d.-c). Tech.Co. Retrieved March 23, 2024, from <https://tech.co/asset-tracking/glpi-review>

GLPi Review 2024: Features, Pros, Cons and Pricing. (n.d.-d). Tech.Co. Retrieved March 25, 2024, from <https://tech.co/asset-tracking/glpi-review>

GLPi Reviews—Pros & Cons, Ratings & more. (n.d.). GetApp. Retrieved March 23, 2024, from <https://www.getapp.com/it-management-software/a/glpi/reviews/>

TheKnowledgeAcademy. (n.d.-a). Prototype Model In Software Engineering: All You Need To Know. Retrieved March 25, 2024, from <https://www.theknowledgeacademy.com/blog/prototype-model-in-software-engineering/>

TheKnowledgeAcademy. (n.d.-b). Prototype Model In Software Engineering: All You Need To Know. Retrieved March 25, 2024, from <https://www.theknowledgeacademy.com/blog/prototype-model-in-software-engineering/>

TheKnowledgeAcademy. (n.d.-c). Prototype Model In Software Engineering: All You Need To Know. Retrieved March 25, 2024, from <https://www.theknowledgeacademy.com/blog/prototype-model-in-software-engineering/>

TheKnowledgeAcademy. (n.d.-d). Prototype Model In Software Engineering: All You Need To Know. Retrieved March 25, 2024, from <https://www.theknowledgeacademy.com/blog/prototype-model-in-software-engineering/>

Untitled.vpd | Visual Paradigm Online. (n.d.). Retrieved March 31, 2024, from <https://online.visual-paradigm.com/w/gjxlgzpl/app/diagrams/#diagram:proj=0&workspace=gjxlgzpl&type=ActivityDiagram&width=11&height=8.5&unit=inch?themeSketch=1>

APPENDIX

Appendix A: Data set

DEPARTEMENT	Utilisateur	Marque	N° Serie
DAD	KUMAR	Hp Laserjet P 2035	VNC4G42514
	KODOCK	Hp COLOR LASERJET PRO MFP 277	
	SALLE	CANON IR 2520	
SECURITY	BUREAU SECURITY MANAG	Hp Deskjet 4530	TH6813D0ZQ
	OPERATING SECURITY	Hp Laserjet Pro MFP 227	VNC4509749
SUSTAINABILITY	AMBASSA ADELAIDE	Hp Deskjet 3000	CN1CE32 H5W
	AGATH	Hp LASERJET PRO M120 a	VNC6K06691
	GUINSOM ROMEO	HP	
	BUREAU ASS SOCIAL INTERNE	Hp Lserjet P2035	VNC4G23083
	MENGUE J	Hp Lserjet P2035	CNCF460717
	NJE NJE	Hp Laserjet M426 DW	PHB8JD10VL
AUDIT INTERNE	YMBNE	Hp Deskjet 3000	CN0AE11GP2
DDP	DDP	Hp Laserjet M806	JPDVL8DOJZ
MAGASIN	NJOH O	Hp Laserjet P2035	VNC4B29741
	TIOBO	Hp Laserjet P2035	VNC4G61153
	NDZANA R	Hp Laserjet P2035	VNC4F52297
	SALLE	CANON IR 2520	(21) RML87537
COMMUNICATION	ZOBO J C	Hp Deskjet 3000	CN1CE32K2Z
DPE/MA	Ex RAJESH	Hp Deskjet 4645	CN530570XR
	SECRETARIAT	CANON IR 2202	
	SECRETARIAT	Hp Laserjet Pro MFP 227	VNC4G22731
	KWEM OSCAR	Hp Officejet Pro 8710	CN6AIH323C
	MINTANGA	Hp Laserjet Pro M402	PHCGD04543
	TSAFFO M	Hp Laserjet Pro M402	
DT	ATSANG L	Hp Laserjet Pro M402	PHC6C08359

Figure 12: dataset

Appendix B: Code Listing

Algorithm to delete equipment from database

Algorithm ProductDelete (request, pk)

1. Begin
2. Check if user is logged in
3. if not, redirect to 'user-login'
4. Check if user role is 'Admin'
5. if not, access is denied
6. Get the product with id 'pk'
7. If the request method is 'POST'
8. Delete the product
9. Redirect to 'dashboard-products'
10. Else
11. Prepare the context with 'item' as the product
12. Render 'dashboard/products_delete.html' with the context
13. End if
14. End

Function to delete equipment from the database

@login_required (login_url='user-login')

@allowed_users (allowed_roles=['Admin'])

Def product_delete (request, pk):

 item = Product.objects.get (id=pk)

```
if request.method == 'POST':  
  
    item.delete ()  
  
    return redirect ('dashboard-products')  
  
context = {  
  
    'item': item  
  
    }  
  
return render (request, 'dashboard/products_delete.html', context)
```

Appendix C: statically analysis and graphical representations

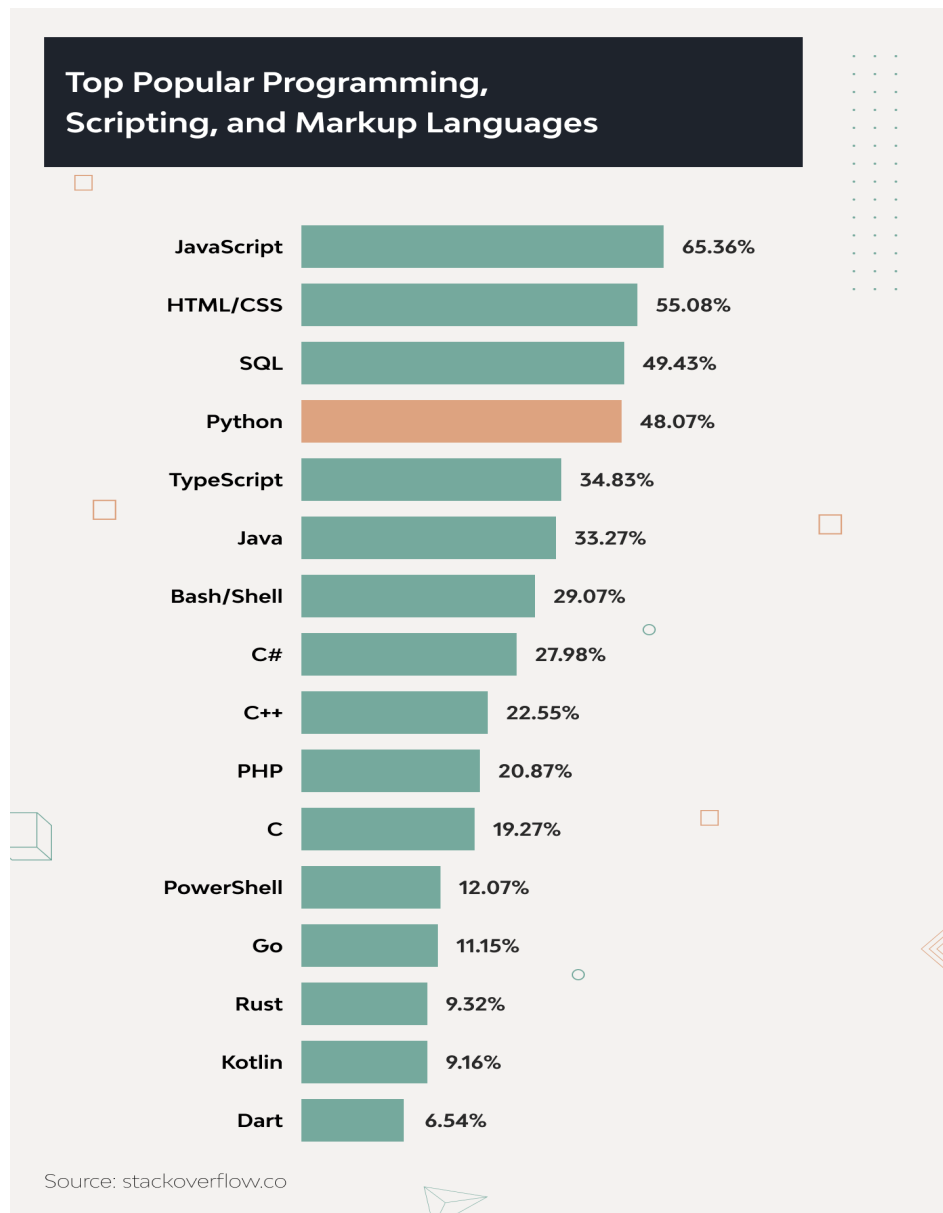


Figure 13

Source

(The Knowledge Academy, n.d.-b)

Appendix D: Certificate of completion of the internship


HEVECAM
CORRIE MACCOLL LIMITED

DIRECTION DES RESSOURCES HUMAINES
.....
SERVICE DEVELOPPEMENT RH & GESTION
DES CARRIERES
.....
NSGA/AZM/BDM

Niété, le 05 Octobre 2023

ATTESTATION DE STAGE N° 48 - DARH/2023

Nous soussignés, Société Anonyme HEVEA-CAMEROUN (HEVECAM S.A) BP 174 KRIBI.

Attestons par la présente que Monsieur TCHINDE LUC KEVIN a effectué un stage ACADEMIQUE à la Direction Financière/Technologie de l'information de notre société, du 20 Juin au 20 Septembre 2023.

En foi de quoi la présente ATTESTATION DE STAGE lui est délivrée pour servir et valoir ce que de droit. /-

Copie à:
- Chrono


HEVECAM S.A.
ALAIN CHRISTIAN BIHOU

Société Anonyme Hévéa Cameroun (HEVECAM S.A)
Share capital: 15 747 950 000 FCFA - Headquarters: Niété-Kribi - Cameroon
P.O BOX 174 KRIBI / 1298 DOUALA - Email: hevecamsa@hevecam.com
Trade Registry: 89 - Tax Payer's Card Number: M077500001799U
Tél: (237) 222 460 736 / 222 460 732

Figure 14 Certificate of completion of the internship

Appendix E: Enterprise evaluation grid

Appendix F: prototype

Figure 16 prototype 1

Figure 17 prototype 2