

REPUBLIQUE OF CAMEROON

Peace-work-fatherland



**AFRICAN INSTITUTE OF COMPUTERS
SCIENCES**

BP: 13 719 Yaoundé (Cameroun)

Tel: (237) 22 72 99 57 / (237) 22 72 99 58

Site Web: www.iaicameroun.com

REPUBLIQUE DU CAMEROUN

Paix -travail-patrie



One Love, One Mind, One Legacy!

linkedIn: shaderl

Tel: (237) 678 358 509 / (237) 692 199 722

Site Web: www.shaderl.com

INTERNSHIP REPORT

THEME : CONCEPTION AND IMPLEMENTATION OF AN AI-ASSISTED AGRIFUTURE INVESTMENT PLATFORM

Carried out from the 1st of July 2025 to the 30th of September 2025 in view
of obtaining a HTD in Software Engineering

Written By :

SOMO REAN GIGGIS KINYUY

LEVEL 2 STUDENT AT AICS
CAMEROON

ACADEMIC SUPERVISOR

Mr. AGBOR Anderson

Lecturer at AICS

PROFESSIONAL SUPERVISOR

Mr. ASANE Derick

C.E.O of SHADERL

ACADEMIC YEAR 2024-2025



DEDICATION

TO THE WIRNKAR'S
FAMILY



ACKNOWLEDGEMENTS

My gratitude goes to the following people who assisted in the realization of this project:

- ❖ The Resident Representative of AICS-Cameroon, Mr. ARMAND CLAUDE ABANDA, for his words of encouragement and the different advices brought to us.
- ❖ The CEO of SHADERL, Mr. ASANE DERICK for the numerous advices and professional expertise shared with us.
- ❖ Mr. AGBOR ANDERSON, our academic supervisor, for a great follow up and advices during the realization of this document.
- ❖ To our academic teachers Mrs. TCHOUTOUO ISABELLE, and Mr. MESSIO ULRICH for their advice and assistance in realizing this document.
- ❖ To all my teachers of **AICS CAMEROON** for the knowledge they impacted me with for me to be where I am today.
- ❖ To my beloved parents MR & MRS WIRNKAR for their endless love, moral and financial support.
- ❖ To my brother ASHERY ALEXENDA and my sister MARY SAMANTA for their love and care regarding my studies.
- ❖ To all our classmate for their support and collaborative work throughout the academic year.



SUMMARY



LIST OF FIGURES



LIST OF TABLES



GLOSSARY



ABSTRACT

Theme: “Design and Implementation of an AI-Assisted AgriFuture Invest Platform”

The theme **AI-Assisted AgriFuture Invest Platform** introduces an innovative, intelligent, and secure digital ecosystem that connects farmers, firms, and investors. This platform is designed to streamline agricultural investment processes while leveraging artificial intelligence to enhance trust, transparency, and decision-making.

Key features include an AI-powered recommendation engine that suggests profitable agricultural projects for investors, fraud detection mechanisms to ensure secure transactions, and AI-assisted proposal generation tools that help firms present their projects in a clear, professional manner. The platform also provides a structured investment workflow, enabling investors not only to fund but also to actively participate in project monitoring and follow-up meetings, ensuring accountability and transparency.

Through an intuitive interface, investors can consult available agricultural projects, analyze their feasibility with AI insights, and proceed with secure wallet-based payments. Farmers benefit from reliable financing and structured project phases, while firms can showcase projects to a broader audience with credibility. The AI chatbot enhances user engagement by guiding participants through processes, answering questions, and offering predictive insights.

This system offers a **comprehensive and interactive experience**, bridging the gap between agricultural stakeholders and investors. By combining AI-driven recommendations, fraud detection, and collaborative investment management, the platform redefines agricultural investment in Cameroon and beyond. It empowers users with data-driven decisions while fostering trust and sustainable growth in the agri-business sector.

Keywords:

- ❖ · AI technology
- ❖ · Agricultural investment
- ❖ · Project proposals
- ❖ · Investors, Firms, Farmers

.



RESUME

Thème : Conception et Mise en œuvre d'une Plateforme d'Investissement AgriFuture Assistée par l'IA

Le thème *Plateforme d'Investissement AgriFuture Assistée par l'IA* introduit un écosystème numérique innovant, intelligent et sécurisé qui connecte les agriculteurs, les entreprises et les investisseurs. Cette plateforme est conçue pour rationaliser les processus d'investissement agricole tout en exploitant l'intelligence artificielle afin de renforcer la confiance, la transparence et la prise de décision.

Les fonctionnalités clés comprennent un moteur de recommandation alimenté par l'IA qui suggère aux investisseurs des projets agricoles rentables, des mécanismes de détection de fraude pour garantir des transactions sécurisées, et des outils de génération de propositions assistés par l'IA qui aident les entreprises à présenter leurs projets de manière claire et professionnelle. La plateforme fournit également un flux de travail structuré pour l'investissement, permettant aux investisseurs non seulement de financer mais aussi de participer activement au suivi et aux réunions de suivi des projets, garantissant ainsi la responsabilité et la transparence.

Grâce à une interface intuitive, les investisseurs peuvent consulter les projets agricoles disponibles, analyser leur faisabilité avec des analyses fournies par l'IA, et procéder à des paiements sécurisés via un portefeuille électronique. Les agriculteurs bénéficient d'un financement fiable et de phases de projet structurées, tandis que les entreprises peuvent présenter leurs projets à un public plus large avec crédibilité. Le chatbot IA améliore l'engagement des utilisateurs en guidant les participants à travers les processus, en répondant aux questions et en offrant des analyses prédictives.

Ce système offre une expérience complète et interactive, comblant le fossé entre les parties prenantes agricoles et les investisseurs. En combinant recommandations basées sur l'IA, détection de fraude et gestion collaborative des investissements, la plateforme redéfinit l'investissement agricole au Cameroun et au-delà. Elle donne aux utilisateurs les moyens de prendre des décisions fondées sur les données tout en favorisant la confiance et la croissance durable dans le secteur agro-industriel.

Mots-clés :

- ❖ · Technologie IA
- ❖ · Investissement agricole
- ❖ · Propositions de projets
- ❖ · Investisseurs, Entreprises, Agriculteurs



GENERAL INTRODUCTION

To bridge the gap between theory and practice, second-year students at **AICS Cameroon** are required to undertake a **three-month academic internship** in an enterprise. This internship enables them to apply the knowledge acquired in class within a professional environment.

For this reason, we applied for an internship at **SHADERL**, a company that specializes in providing software solutions. During our stay at SHADERL, we were assigned the project theme: **“Conception And Implimentation Of An AI-Assisted Agrifuture Investment Platform.”**

This report documents the entire process of the internship and the development of the AgriFuture Investment Platform. The platform is aimed at revolutionizing the agricultural investment ecosystem by connecting farmers, firms, and investors through a transparent and interactive digital solution. We divided this report into 2 parts which is as follows:

PART I: The Insertion phase: This book presents the company in which we spent our internship period and the integration of interns

PART II: Technical phase: which is composed of the following

- ❖ **PHASE I:** Existing System: Here, we shall present the already present system in place, that is the one used for consultation and follow-up purposes.
- ❖ **PHASE II:** The specification book: In this book, we specify the needs of the user taking into considerations the time and cost of the project.
- ❖ **PHASE III:** Analysis book: Here, we present the analysis method chosen with the presentation of all the diagrams used for the analysis of this project.
- ❖ **PHASE IV:** Design phase: This presents the generic and detailed conception of the project to bring out real world constituents.
- ❖ **PHASE V:** Deployment phase: This phase will permit us to visualize the implementation process of the solution
- ❖ **PHASE VI:** Test of functionalities: In this phase, we shall present to you the different functionalities or modules of our application and how they work.
- ❖ **PHASE VII:** user guide: which guide the user on how to install and use the application



PART I : INSERTION PHASE



Preamble

This section of the report will cover details of how we were welcomed in the host company, presentation, organization and brief introduction to our project.

Content

INTRODUCTION

- I. WELCOME AND INSERTION
- II. GENERAL PRESENTATION OF THE COMPANY
- III. ORGANIZATION OF THE COMPANY
- IV. GEOGRAPHICAL LOCATION
- V. BRIEF PRESENTATION OF PROJECT THEME

CONCLUSION



INTRODUCTION

The insertion phase in a company is a period during which we discover our working environment, the staff of the company and other interns. Here, we will begin by discussing our first two weeks in the company, how we were welcomed and how we began adapting to our internship environment, we will proceed by exploring the history of the company, discovering its missions, learning what its major activities are, and witnessing some of its key realizations. We will also get to understand how the company is structured administratively and functionally so that it operates effectively and accomplishes its goals. Furthermore, we will look at the hardware equipment used by the company, and the software resources used in its daily operations. We will then introduce our chosen theme for the internship period, briefly elaborate on It.



I-WELCOME AND INSERTION

We arrived at SHADERL on Tuesday 01st July 2025, at 08:00 a.m. we were received by the shaderl team who introduced us to our workspace, gave us an official welcome to the enterprise, its activities and its different rules and regulations. Also, we discussed on the enterprise's preferred languages and frameworks, our professional supervisor encouraged us through his past working experiences in different enterprises.

The first two week we started with our internship report by configuring our MS WORD processor, reviewing key concepts like page numbering, styles, section breaks, page breaks etc. There was a talk on project ideas, we were advised to propose project ideas and we were assigned the tasks to carry out research on these ideas. The objective was that the enterprise did not want to impose some themes on us. For those who could not find a theme, the enterprise could propose an idea for them to think about.

II-GENERAL REPRESENTATION OF SHADERL

A-Background:

Shaderl is a dynamic, non-governmental tech startup founded in 2023 by **Asane Derick** with a bold vision: to bridge the gap between traditional industries and the rapidly evolving digital economy. Recognizing the urgent need for digital transformation, Shaderl is committed to providing **cutting-edge IT solutions** while empowering the next generation of innovators.

B- Mission

At Shaderl, we harness the power of **AI and cutting-edge technology** to democratize digital transformation. Our mission is to **empower individuals and businesses** with intelligent tools, accessible education, and personalized mentorship bridging the gap between today's potential and tomorrow's opportunities. We believe AI should **augment human potential, not replace it**. By making AI-driven solutions practical and ethical, we equip the next generation to lead, innovate, and thrive in an era where **human creativity meets machine intelligence**.

C-Vision

At Shaderl, we envision a world where **technology unites humanity**, where innovation is driven by collective purpose, not just profit. By putting **"PeopleFirst"**, we strive to build a future where:



CONCEPTION AND IMPLEMENTATION OF AN AI-ASSISTED AGRIFUTURE INVESTMENT PLATFORM



One Love→ Technology fosters **inclusion, empathy, and global collaboration**, breaking down barriers.

One Mind→ AI and digital tools **amplify human potential**, creating shared knowledge and opportunity.

One Legacy→ Every individual we empower leaves a lasting impact, shaping a **smarter, kinder, and more connected world**.

We don't just adapt to the digital age; we **redefine it with humanity at the core**.

D- ACTIVITIES

The activities of SHADERL range from computer sciences, engineering, and training. We can outline the following:

- Conception realization, and hosting of websites.
- Software development and maintenance.
- Training in Software related fields.
- Conception and realization of multimedia.
- IT consulting and innovation.
- IT support.

III-ORGANISATION OF SHADERL

Shaderl is administratively organized as follows:

a- Executive Leadership

This department is responsible for:

- ✓ Setting the company's vision, mission, and strategic direction
- ✓ Making high-level decisions about investments, partnerships, and growth
- ✓ Representing Shaderl in key meetings with investors and government agencies
- ✓ Ensuring all departments align with the company's core values and objectives
- ✓ Overseeing the overall performance and sustainability of the organization

b. Software Engineering Department

This department is responsible for:

Written by : Somo Rean Giggis Kinyuy,
a 2nd year student studying Software Engineering at AICS Cameroon, Paul Biya Technological Center of Excellence, for the **Academic year 2024–2025**



CONCEPTION AND IMPLEMENTATION OF AN AI-ASSISTED AGRIFUTURE INVESTMENT PLATFORM



- ✓ Designing, developing, and maintaining all of Shaderl's software products
- ✓ Implementing AI and machine learning solutions for company offerings
- ✓ Ensuring software security, scalability, and optimal performance
- ✓ Collaborating with other departments to understand technical requirements
- ✓ Staying updated with emerging technologies and industry best practices

c. Human Resource Department

This department is responsible for:

- ✓ Recruiting and onboarding top talent that aligns with Shaderl's values
- ✓ Managing employee relations, welfare, and performance evaluations
- ✓ Developing training programs to enhance staff skills and capabilities
- ✓ Maintaining company culture and ensuring a positive work environment
- ✓ Handling compensation, benefits, and conflict resolution

d Communication Department

This department is responsible for:

- ✓ Managing Shaderl's public image and brand reputation
- ✓ Developing and executing marketing and PR strategies
- ✓ Handling all internal and external communications
- ✓ Managing social media platforms and digital content
- ✓ Organizing corporate events and press engagements

e Department of Financial Affairs

This department is responsible for:

- ✓ Managing all financial operations and accounting
- ✓ Preparing budgets and financial forecasts
- ✓ Handling payroll, taxes, and financial reporting
- ✓ Ensuring compliance with financial regulations
- ✓ Managing investments and financial partnerships



B- Functional organisation

The functional branch of shaderl is organized as follows:

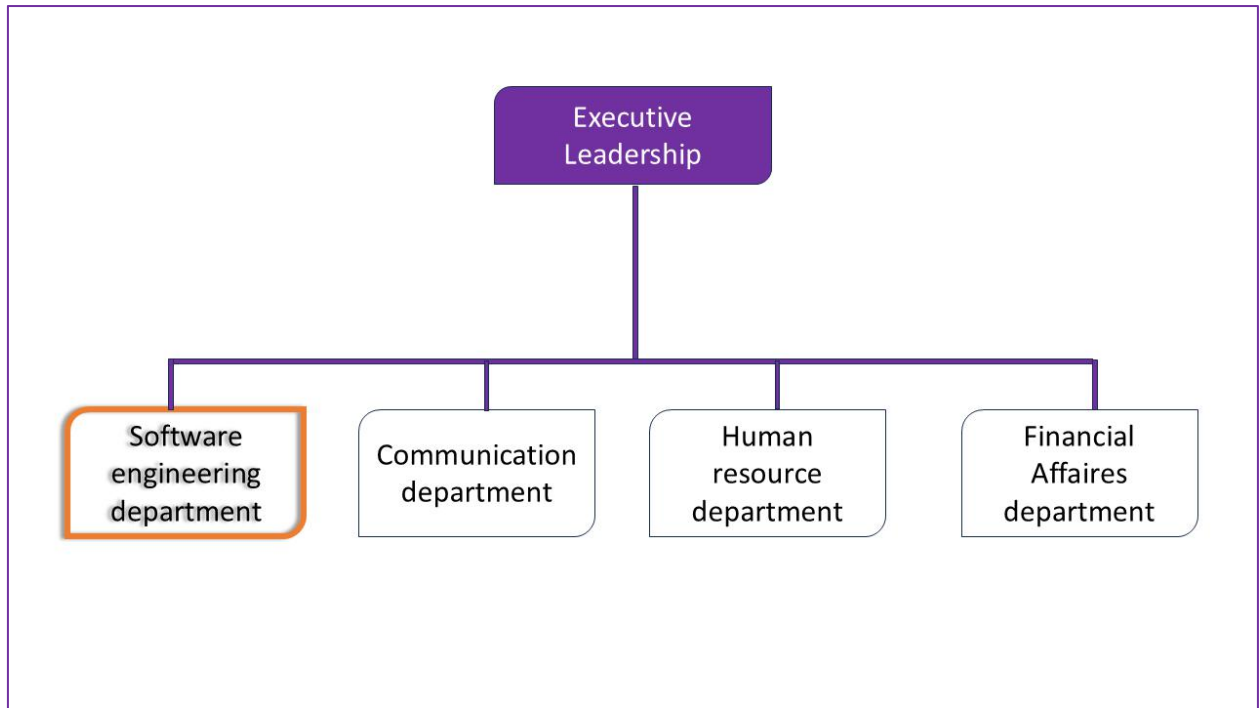


Figure 1



IV- GEOGRAPHICAL LOCATION

1. History

The **SHADERL** project was born from a spark of genius during the third-year engineering internship of **Mr. SANAMA Léopold Armel** at **IAI-Cameroon**. The idea matured into a real-world venture, and in **February 2025**, it officially became a registered entity with **SARL (Limited Liability Company)** status.

In its early days, SHADERL focused on training young students from universities and professional institutions, equipping them with practical skills to build high-quality software products for the Cameroonian society. Over time, SHADERL gained recognition for its commitment to youth empowerment, innovation, and software excellence.

Today, SHADERL partners with numerous young professionals and continues to spearhead diverse technology-driven projects across Cameroon.

2. Location

SHADERL is located in **Yaoundé**, near **Marie Albert**. The company's premises are easily accessible by road, and the map below illustrates its exact location for ease of navigation:

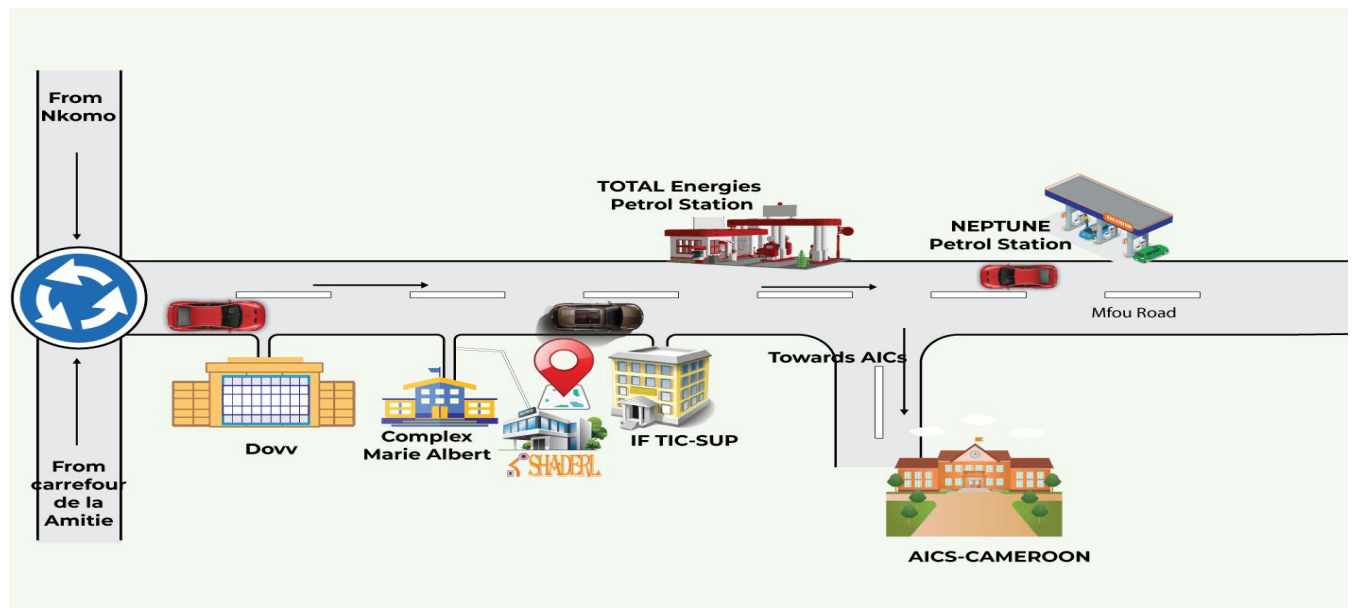


Figure 2



V- BRIEF PRESENTATION OF THE PROJECT THEME

Upon our arrival at **SHADERL**, we were encouraged to propose potential project ideas for our internship. In line with SHADERL's mission to actively contribute to the development of innovative software solutions, we identified an opportunity to explore our creativity and proposed the theme: **“Design and Implementation of an AgriFuture Investment Platform.”**

This platform aims to bridge the gap between firms and potential investors by providing a transparent, secure, and user-friendly investment system tailored to the agricultural sector. The system is composed of several modules:

Admin Dashboard: Responsible for the management of platform users, including the creation of investor and farmer accounts, approval of project listings, and system-wide settings.

Investor Dashboard: Allows investors to view agricultural projects, access detailed proposals, track investment performance, and receive updates or reports from farmers.

Farmer Dashboard: Enables farmers and agricultural startups to register their projects, upload funding proposals, share updates, and interact with potential investors.

Public Home Interface: Offers general visitors the ability to browse featured agricultural projects, view statistical insights, and learn about investment opportunities before signing up. Through this platform, we aim to support digital transformation in agriculture and empower both farmers and investors with a reliable tool for funding and development.



CONCLUSION

Our insertion phase at **SHADERL** lasted for a period of **two (2) weeks**. During this time, we familiarized ourselves with the company's working environment, which was well-structured and equipped with the necessary tools to promote hard work, productivity, and collaboration.

We took time to understand the **mission** of the company and its various **ongoing activities**, which revolve around the development of innovative software solutions tailored to societal needs. We were also introduced to other interns with whom we engaged in collaborative tasks, fostering teamwork and knowledge exchange.

At the end of this orientation period, we collectively decided on our project theme: **“Design and Implementation of an AI-Assisted AgriFuture Investment Platform.”** This marked the beginning of our practical journey toward building a digital solution aimed at revolutionizing investment in the agricultural sector.



PART II : TECHNICAL PHASE



Preamble

The technical phase is a document that describes the process, progress, or results of a scientific research. It also includes a detailed analysis and the conclusions drawn from that research.

Content

PHASE I: THE EXISTING SYSTEM

PHASE II: THE SPECIFICATION BOOK

PHASE III: THE ANALYSIS PHASE

PHASE IV: CONCEPTION PHASE

PHASE V: REALIZATION PHASE

PHASE VI: TEST OF FUNCTIONALITIES

PHASE VII: INSTALLATION GUIDE AND USER GUIDE



PHASE I : THE EXISTING SYSTEM



Preamble

The existing system is a document that provides a view of how the current system works, how it carries out its different activities, it equally provides a deep understanding of the system associated to the various limitations and the problems that result from it and we propose the solution.

Content

TRODUCTION

- I. PRESENTATION OF THE PROJECT THEME
- II. STUDY OF THE EXISTING SYSTEM
- III. CRITICISMS OF THE EXISTING SYSTEM
- IV. PROBLEMATICS
- V. PROPOSED SOLUTION



INTRODUCTION

As Engineers our role in the society is problem-solving which is targeted at making life easier for everyone and in order to bring forth a solution, we ought to ask ourselves some questions to know why the problem exist in the first place and how it is being dealt with at a time. The existing system refers to the system put in place to carry out the work done in the field on which our theme is based. Understanding this system is a great step in solving the problems that we might identify. It's of great importance that we take into consideration this system before proposing a suitable solution that will ameliorate or replace the current one.



I. Presentation of the Project Theme

In a world where agriculture remains a vital pillar of economic development especially in Cameroon smallholder farmers, agricultural startups, and agribusinesses often struggle to access sufficient funding and visibility. At the same time, many investors are seeking reliable and impactful opportunities to support sustainable ventures. Bridging this gap is essential for national development, food security, and youth engagement in agriculture.

To address this need, we proposed the project theme:

“Design and Implementation of an AI-Assisted AgriFuture Investment Platform.” This platform is intended to serve as a **digital bridge between agricultural entrepreneurs and potential investors**. It is a web-based application that will enable farmers and agribusiness project owners to showcase their initiatives, submit investment proposals, and connect with interested funders.

The main objectives of the platform include:

- Creating a **centralized database** of verified agricultural projects seeking investment.
- Allowing investors to **browse, assess, and fund** promising agricultural initiatives
- Providing **transparency and accountability** through project tracking, reporting, and communication tools.
- Supporting **economic inclusion** by giving access to funding for rural and emerging farmers who often face financing barriers.

In situations where a project owner requires guidance on how to present or manage a project for investment, the platform will also provide access to **advisors or agri-orientators** professionals who can help shape business models, improve proposals, and recommend pathways for success.

By fostering collaboration between agriculture and technology, the AgriFuture Investment Platform will help drive innovation, reduce investment risk, and promote the growth of sustainable agribusiness in Cameroon and beyond.



II. Description of the Existing System

In the current agricultural investment ecosystem, the connection between investors and agricultural entrepreneurs in Cameroon remains weak, informal, and largely inefficient. Traditional methods used by farmers and agribusiness startups to seek funding include physical visits to financial institutions, informal requests through personal networks, and occasional presentations at agricultural fairs or workshops. These methods are **time-consuming, geographically limited, and often lack transparency and structure.**

On the other hand, investors interested in supporting agricultural ventures often face challenges such as:

- **Limited access to reliable project proposals**
- **Lack of credible platforms to evaluate risks**
- **No centralized system for follow-up and reporting**
- **Concerns over the accountability and traceability of their funds**

Currently, there is **no dedicated digital platform** that brings together investors and agricultural project owners in a structured, interactive, and secure manner. Most agricultural financing platforms that exist either target large agribusinesses or focus on loan-based financing, excluding smallholder farmers and startups from the ecosystem.

Moreover, farmers in rural areas often lack the technological tools and support systems to promote their projects to potential sponsors or partners. This results in a **missed opportunity for agricultural innovation, job creation, and sustainable food production.**

The absence of a centralized digital solution limits the growth of agriculture as a business sector, discourages private investment, and restricts youth participation in agritech innovations.

Therefore, there is a critical need for a modern solution that enables **visibility, connectivity, accountability, and financial flow** between investors and project owners hence the motivation for the **AI-Assisted AgriFuture Investment Platform.**



III. Criticism of the Existing System

After collecting and analyzing information regarding existing systems for agricultural investment and support, we identified several **key weaknesses** that hinder their effectiveness. These limitations have practical consequences for both farmers and investors, and if not addressed, they will continue to slow down progress in the agricultural sector.

The table below presents a summary of the identified **limitations**, their **consequences**, and **proposed solutions**:

I. Limitations, Consequences, and Solutions of the Existing System

Table 1

| Limitations | Consequences | Proposed Solutions |
|---|---|---|
| Lack of accessible information on viable projects | - Investors unaware of opportunities- Projects remain underfunded | Create a platform that lists and showcases verified agricultural projects |
| Absence of professional investment advisors | - Investors lack guidance- Farmers struggle to prepare proper proposals | Integrate expert advisors or “agri-orientators” to support both farmers and investors |
| No centralized concours or grant notifications | - Missed opportunities- Mismatch between project goals and funding rounds | Implement notification boards for upcoming grant opportunities and government support |
| Poor decision-making tools for investors | - Risk of funding low-impact or unfeasible projects | Include performance analytics, success metrics, and review systems in the platform |



CONCEPTION AND IMPLEMENTATION OF AN AI-ASSISTED AGRIFUTURE INVESTMENT PLATFORM



| Limitations | Consequences | Proposed Solutions |
|---------------------------------------|--|---|
| Farmers unaware of required standards | - Project rejection due to poor planning or presentation | Provide templates, guidelines, and training materials on the platform |



IV. Problematic

Agriculture is a major pillar of economic development in Cameroon, yet many smallholder farmers and agricultural startups face serious challenges in accessing funding to grow their businesses. On the other hand, investors both local and international are often unable to find trustworthy and promising agricultural ventures to support.

This leads us to ask the core question:

"How can we facilitate the connection between agricultural entrepreneurs and potential investors while ensuring transparency, visibility, and accountability in the investment process?"

The **AgriFuture Investment Platform** seeks to provide a reliable, user-friendly, and secure digital space to address this issue.



V. Proposed Solution

Following our analysis and criticism of the existing system, we propose to **design and implement a web-based platform** that will:

- ✓ **Connect agricultural project owners with investors**, enabling them to showcase their ideas and request funding.
- ✓ **Provide a searchable catalog of verified agricultural projects**, with detailed descriptions, goals, locations, and funding needs.
- ✓ **Offer a secure and transparent investment system**, where investors can track the progress of the projects they support.
- ✓ **Include advisor support (Agri-orientators)** to guide farmers in preparing strong investment proposals and improving project feasibility.
- ✓ **Integrate a notification board** to inform users about upcoming agricultural grants, concours, and relevant events.
- ✓ **Enable geographic mapping** of projects and farm locations to improve visibility and traceability.
- ✓ **Support direct communication** between project owners and investors through in-app messaging or contact requests.

Through this platform, we aim to **boost agricultural innovation, encourage private investment**, and support the **digital transformation of the agricultural sector in Cameroon**.



Conclusion

Having reached the end of this part of our report, we conducted a thorough analysis of the existing system through research and observations in the agricultural sector. Our findings revealed significant limitations, including lack of access to funding opportunities, poor visibility of promising agricultural projects, and a general absence of structured platforms to connect investors with project owners.

These limitations result in underfunded initiatives, missed opportunities, and slowed progress in agricultural innovation and development. Based on these findings, we proposed practical solutions aimed at addressing each of these issues.

Finally, we introduced our project solution: the **Design and Implementation of an AI Assisted AgriFuture Investment Platform** a web-based application that bridges the gap between firms and investors. The platform is designed to foster collaboration, increase transparency, and enhance the agricultural investment ecosystem in Cameroon.



PHASE II : SPECIFICATION BOOK



Preamble

The Specification book is a document which is been established by the customer needing a particular product and the producer (engineer) who is to create the product, which is to be respected to its fullness during the realization of the product. From this, we obtain the major importance of the specification book.

Content

INTRODUCTION

I. CONTEXT AND JUSTIFICATION

II. OBJECTIVES OF THE PROJECT

III. EXPRESSION OF NEEDS

IV. PROJECT PLANNING

V. ESTIMATED COST OF THE PROJECT

VI. PROJECT CONSTRAINTS

VII.DELIVERABLES

CONCLUSION



INTRODUCTION

The specification book of our reports helps us provide details about our theme, to improve our understanding of it and increase the likelihood of it succeeding. To delimitate the scope of our project, we will specify the context of our theme. From the context, we will list the problems we have identified in our context and that we have decided to address throughout the project. After presenting our solution, we will talk about the objective we have set for the project. Also, we will explore the needs to which our system will respond both at the functional and non-functional level. We will then look at the estimated financial requirements for our project, and establish a plan we will follow to complete our project on time. From here we will discuss what is expected of us by the end of the project under the project deliverables.



I. CONTEXT AND JUSTIFICATION

A. Context

The agricultural sector plays a vital role in the economic development of Cameroon, employing a large portion of the population and contributing significantly to food security and national income. Despite its importance, the sector continues to face major challenges one of the most critical being **limited access to investment opportunities** for smallholder farmers, agricultural startups, and rural entrepreneurs.

At the same time, many **potential investors** both individuals and institutions are seeking high-impact, sustainable ventures to support, but lack a **reliable and centralized platform** through which they can identify and evaluate viable agricultural projects.

In this context, the **design and implementation of a web-based agricultural investment platform** becomes not just relevant, but essential. The **AgriFuture Investment Platform** aims to bridge this gap by creating a **secure, interactive, and accessible digital space** where agricultural project owners can present their ideas and connect with potential investors across Cameroon and beyond.

B. Justification

The justification for this project lies in the **inefficiencies and inequities** in the current agricultural financing landscape. Many farmers and agripreneurs lack the tools, exposure, and technical know-how to attract investment or present their projects professionally. Conversely, investors are often unable to discover reliable projects or assess their feasibility due to the absence of standardized presentation platforms.

This platform will:

- ◆ Promote **financial inclusion** by giving small-scale farmers access to a broader investor base.
- ◆ Encourage **agricultural innovation** by funding projects that would otherwise remain invisible.
- ◆ Increase **transparency and trust** in the investment process through project tracking, documentation, and communication tools.
- ◆ Support national goals around **youth employment, digital transformation, and sustainable agriculture**.

The **AgriFuture Investment Platform** is, therefore, not only a technological solution but also a strategic tool for empowering the next generation of agricultural leaders and transforming the way agricultural investments are made in Cameroon.



II. OBJECTIVES OF THE PROJECT

A. General Objective

The main objective of this project is to **design and implement a centralized web-based platform** that facilitates **investment in the agricultural sector** by connecting agricultural project owners with potential investors. This platform aims to provide a transparent, secure, and accessible environment where both parties can interact, share resources, and build mutually beneficial partnerships.

B. Specific Objectives

- ✓ **To enable farmers and agripreneurs** to register and showcase their agricultural projects with detailed information, including funding needs, location, and impact goals.
- ✓ **To provide investors** with a structured catalog of verified agricultural projects, allowing them to browse, filter, and support the initiatives that align with their interests.
- ✓ **To allow administrators** to manage platform users (farmers and investors), approve project listings, and monitor platform activity for quality control and transparency.
- ✓ **To support agri-orientators or advisors**, who will guide project owners in presenting their proposals effectively and assist investors in making informed decisions.
- ✓ **To integrate a notification board** that regularly updates users on grant opportunities, agricultural events, government concours, and other sector-related news.
- ✓ **To include communication tools** such as messaging or appointment booking, allowing real-time interaction between investors, farmers, and advisors.



III. EXPRESSION OF NEEDS

1. Functional Needs

Functional needs describe the actions and capabilities available to different users of the system. For the **AgriFuture Investment Platform**, these needs include the following:

System Administrator

- ❖ **Create Accounts:** Responsible for creating accounts for investors, farmers, and advisors (agri-orientators).
- ❖ **Manage Accounts:** Ability to activate, suspend, delete, or block user accounts to maintain platform security and quality.
- ❖ **User Authentication:** Enable users to register and securely log into the platform.

Farmers / Agricultural Project Owners

- **Register and Manage Projects:** Ability to create, update, and manage detailed profiles of their agricultural projects including funding needs, goals, and location.
- **Edit Profiles:** Update personal and project information to facilitate communication with investors.
- **Communicate:** Send and receive messages to and from potential investors or advisors.

Investors

- **Browse Projects:** Search and filter agricultural projects based on criteria such as location, crop type, funding amount, and risk level.
- **Invest & Track:** Initiate investments and track project progress and reports through the platform.
- **Communicate:** Connect with project owners or advisors for further inquiries.

Advisors (Agri-Orientators)

- **Edit Profiles:** Add relevant information and credentials to facilitate trust and communication.
- **Support Projects:** Review project proposals, provide recommendations, and validate farmer submissions.
- **Manage Appointments:** View, accept, or decline consultation requests from farmers or investors.



2. Non-Functional Needs

Non-functional requirements define the overall qualities, characteristics, and constraints of the software system, focusing on **how** the system operates rather than specific functions.

Performance

The platform should respond promptly to user actions, ensuring minimal waiting time even under high user load or simultaneous transactions. Fast response times will enhance user experience and engagement.

Reliability and Availability

- The application must ensure **high uptime** with minimal downtime to provide consistent and uninterrupted access to users.
- Robust error handling and recovery mechanisms should be in place to maintain platform stability.

Usability

- The platform should feature a **user-friendly interface (UI)** that is intuitive and easy to navigate for all users, including farmers, investors, and advisors.
- Accessibility considerations should be integrated to accommodate users with varying levels of technical proficiency.

Maintainability

- The codebase should be clear, modular, and well-documented to facilitate easy future development, debugging, and enhancement.

Security

- The platform must implement strong security measures to protect sensitive user data and financial transactions.

Key security principles to uphold include:

- **Confidentiality:** Ensuring user data and communications are private.
- **Integrity:** Protecting data from unauthorized modification.
- **Authentication:** Verifying the identity of users to prevent unauthorized access.

These non-functional needs are critical to building a trusted, efficient, and scalable platform that supports the growing agricultural investment ecosystem.



IV. PROJECT PLANNING

a. Chronogram of Activities

Table 2

| Phase | Objective | Output | Duration Period |
|---------------------|---|------------------------------|--|
| Insertion | Collecting information about the enterprise | Insertion report/book | 2 weeks (2 th to 15 nd July 2025) |
| Existing System | Study of the existing system | Existing System Analysis | 1 week (16 nd to 22 th July 2025) |
| Specification Book | Specification of user needs | Specification Document | 5 days (23 th to 27 th July 2025) |
| Analysis | System analysis | Analysis Report | 2 weeks (28 th July to 10 th August 2025) |
| Conception | Hardware and software design | Conception Document | 2 weeks (11 th to 24 th August 2025) |
| Realization | System development | Developed System | 3 weeks (25 th August to 14 th September 2025) |
| Testing & Debugging | System testing and debugging | Tested and Debugged Software | 1 week (15 th to 27 th September 2025) |
| Writing User Guide | Creation of user instruction manual | User Guide Document | 1 week (15 th to 27 th September 2025) |
| Deployment | Capture of needs, use case and textual description modeling | Realization Document | 1 week (28 th to 30 th September 2025) |



b. Gantt Project Planning



V. Estimated Cost of Project

a. Software Resources

Table 3

| Resource Designation | Usage | Quantity | Unit Cost (FCFA) | Type |
|-----------------------|----------------------------------|----------|------------------|--------------------------------|
| Microsoft Office | Writing report | 1 | 38,750 | Licensed |
| GitHub | Saving report and source code | 1 | Free (Freemium) | Cloud Storage |
| Google Chrome | Research and viewing application | 1 | Free | Freeware |
| Visual Studio | Writing application code | 1 | Free | Freeware |
| EdrawMax | Drawing geographical location | 1 | Free (Freemium) | Illustrating tool |
| Sybase Power AMC | UML analysis and diagrams | 1 | Free (Freemium) | Modeling tool |
| GanttProject | Building Gantt charts | 1 | Free (Freemium) | Project planning |
| Node Js | Writing backend code | 1 | Free (Freemium) | JavaScript runtime environment |
| NoSQL (e.g., MongoDB) | Storing information (database) | 1 | Free (Freemium) | Database |
| TOTAL 1 | | | 38,750 | |

Written by : Somo Rean Giggis Kinyuy,

a 2nd year student studying Software Engineering at AICS Cameroon, Paul Biya Technological Center of Excellence, for the Academic year 2024–2025



b. Hardware Resources

(Source: ubuy.cm)

Table 4

| Resource | Usage | Quantity | Unit Cost (FCFA) |
|--------------------------|------------------------------------|----------|------------------|
| Desktop (Model: H6AP3S0) | Drafting reports and coding | 1 | 120,000 |
| Printer HP | Printing the report | 1 | 65,000 |
| Modem (TP-Link) | Network connectivity | 1 | 25,000 |
| USB Key (64GB) | Software installation and sharing | 1 | 7,000 |
| MTN Home Box | Collaborative network connectivity | 1 | 15,000 |
| TOTAL2 | | | 232,000 |



c. Human Resources

(Source: paylab.com/cm/)

Table 5

| Role | Number of Days | Quantity | Cost per Day (FCFA) | Total Price (FCFA) |
|------------------|----------------|----------|---------------------|--------------------|
| Project Manager | 91 | 1 | 40,000 | 3,640,000 |
| Analyst | 14 | 1 | 30,000 | 420,000 |
| Designer | 14 | 1 | 20,000 | 280,000 |
| Programmer | 21 | 1 | 15,000 | 315,000 |
| Tester | 13 | 1 | 10,000 | 130,000 |
| Unforeseen Costs | — | — | — | 1,000,000 |
| Total | | | | 5,655,000 |

d. Global Estimation

Table 6

| Category | Total Cost (FCFA) |
|--------------------|-----------------------|
| Software Resources | 38,750 |
| Hardware Resources | 232,000 |
| Human Resources | 5,655,000 |
| Overall Total | 5,925,750 ~ 6,000,000 |
| | Six million. |



VI. CONSTRAINTS

1. Technical Constraint

For the development of the AgriFuture Investment Platform, we have chosen sufficiently robust tools and technologies to ensure a minimum level of security, extensibility, and excellent scalability. During the programming phase, adherence to technical standards will be essential to achieve optimal performance with reduced execution times. Therefore, selecting appropriate development technologies is critical to the project's success.

2. Time Constraint

The entire project is planned to be completed within a period of **13 weeks (3 months)**, starting from the project commencement date of **28th July 2024**. This limited timeframe requires effective time management and adherence to the project schedule.

3. Cost Constraint

The estimated cost of the project is approximately **seven million FCFA**, which may be challenging for the developer (student) to fully fund without external support or sponsorship.

4. Integration Constraints

The system may rely on third-party APIs or external services that impose usage restrictions or present potential performance limitations. These dependencies could affect the platform's functionality and must be managed carefully to ensure smooth integration.



VII. LIST OF PARTICIPANTS AND DELIVERABLES

A. List of Participants

Table 7

| Name | Function | Role |
|----------------------------|------------------------------|------------------------------------|
| Mr. AE DERICK | C.E.O SHADERL | Professional Supervisor at SHADERL |
| Mrs. TCHOUTOUO Isabelle | Lecturer at AICS Cameroon | Academic Supervisor |
| SOMO REAN GIGGIS KINYUY | Intern at SHADERL | Student at AICS Cameroon |

B. Deliverables

In project management, any tangible or intangible output that materializes the result of a task or service is referred to as a **deliverable**. For our project, the key deliverables include:

The Application: The fully developed and functional AgriFuture Investment Platform, ready for deployment and use.

The User Guide: A comprehensive manual providing instructions on how to use the application and its features effectively.

The PowerPoint Presentation: A summary presentation highlighting the project objectives, methodology, results, and conclusions for demonstration purposes.



Conclusion

Having reached the end of this part, the project objectives have been clearly enumerated and defined. The specification book has allowed us to identify the various stakeholders involved in the project, outline the system requirements, and establish a provisional plan necessary for the successful completion of the project.

We will now proceed to the next phase, which is the **analysis phase**. During this phase, we will model our system using a suitable modeling language and unified process methodology. Additionally, we will conduct a comparative study between UML (Unified Modeling Language) and Merise to select the most appropriate approach for our system design.



PHASE III: THE ANALYSIS PHASE



Preamble

The main objective of the analysis phase is to capture the user's need, the delimitation of the field of study and to have a clear understanding of the system in study. To achieve this, we will use UML (Unified Modelling Language) with the 2TUP (2 Track Unified Process) as method applied to UML to analyze the system. We will start by doing a comparison between UML and MERISE, of various unified processes and lastly, we will present the modelling of the solution we propose (Our system)

Content

INTRODUCTION

- I. METHODOLOGY
 - A. COMPARATIVE STUDY OF UML AND MERISE.
 - B. COMPARATIVE STUDY OF UNIFIED PROCESSES.
- II. CHOICE OF THE ANALYSIS METHOD
- III. MODELLING OF THE PROPOSED SOLUTION

CONCLUSION



INTRODUCTION

System development can be viewed as comprising two main components: system analysis and system design. Both are essential for understanding the intricacies of the existing system or the new system being developed. Typically, the analysis and design of information systems aim to facilitate the creation of databases that accurately represent the realities of the studied domain, necessitating the use of a structured design method. Consequently, we have chosen the UML method, as it provides significant advantages for developers seeking a user-centered approach and broad design capabilities. This section of the report will include a comparative study of UML and MERISE, an exploration of unified processes, and an overview of various diagrams that address functional requirements.



I. METHODOLOGY

A. COMPARATIVE STUDY OF UML AND MERISE

i. MERISE

MERISE stands for “Méthode d’Etude et de Réalisation Informatique pour des Systèmes d’Entreprise”. Although it is prescriptive to some extent, MERISE permits the participation of end users and senior management as well as data processing professionals in its decision cycle. MERISE is a method for designing, developing and carrying out IT projects. The goal of this method is to achieve the design of an information system. The MERISE method is based on the separation of data and processing to be carried out in several conceptual and physical models. The essentials of the approach lie in its three cycles: the decision cycle, the life cycle and the abstraction cycle, which cover data and process elements equally. The separation of data and processing ensures longevity in model. Indeed, the arrangement of data does not have to be often overhauled, while treatments are more frequently.

ii. UML

UML (Unified Modelling Language) is a standard notation for the modelling of real-world objects as a first step in developing an object-oriented design methodology. Its notation is derived from and unifies the notations of three object-oriented design and analysis methodologies: Grady Booch's methodology for describing a set of objects and their relationships, James Rumbaugh's Object-Modelling Technique (OMT), Ivar Jacobson's approach which includes a use case methodology. Other ideas also contributed to UML, which was the result of a work effort by Booch, Rumbaugh, Jacobson, and others to combine their ideas, working under the sponsorship of Rational Software. UML captures information about the static and dynamic view of a system. UML 2.5 comprises of 14 diagrams which represent the different views of a system. The 14 diagrams can be subdivided into two, Static or structural and Dynamic diagrams. These diagrams include;

a. STATIC OR STRUCTURAL DIAGRAMS

- ❖ Class diagram
- ❖ Object diagram



- ❖ Component diagram
- ❖ Deployment diagram
- ❖ Composite Structure diagram
- ❖ Package diagram
- ❖ Profile Diagram

ii. BEHAVIOURAL OR DYNAMIC DIAGRAMS

- ❖ Use case diagrams
- ❖ Activity diagram
- ❖ State machine diagram
- ❖ Sequence diagram
- ❖ Communication diagram
- ❖ Global Interaction diagram
- ❖ Timing Diagram

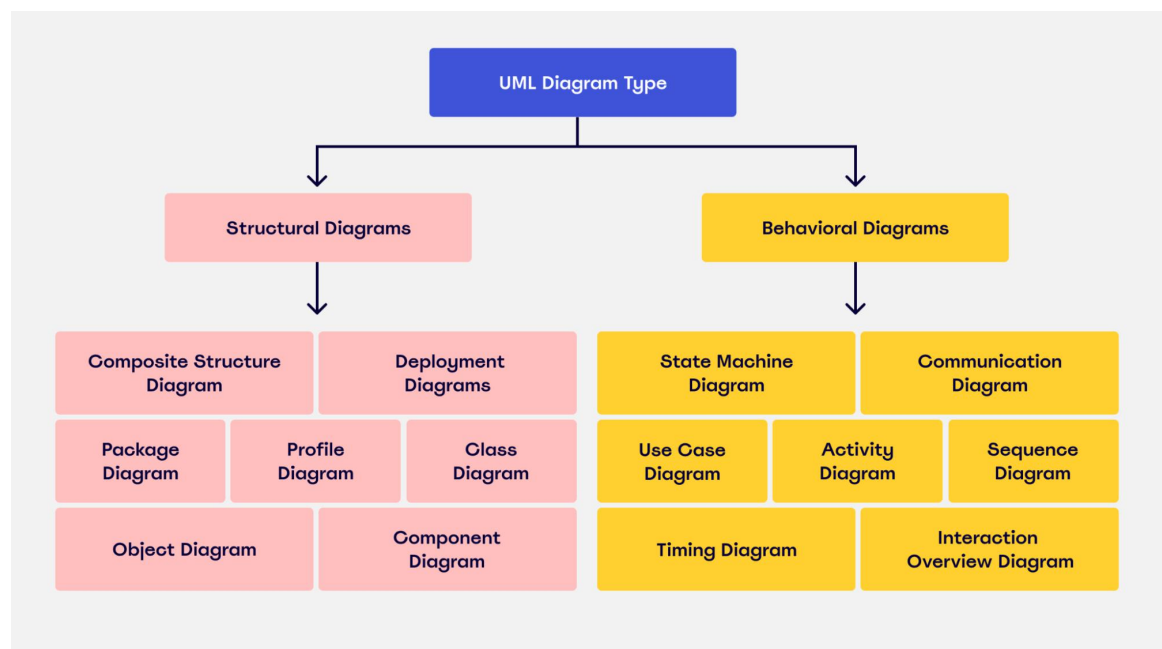


Figure 3

Reference from [Miro](#)

On very important notice is that UML is not a method but a modelling language. As such to give it an approach we need to associate UML to a Unified Process (UP) in order to give our conception a methodology to follow. There exist several Unified



Processes, but our modelling approach will be the 2TUP (Two-track unified process) which we will use in the course of our project.

Table 8

| MERISE | UML |
|--|--|
| It stands for Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprises | Unified Modeling Language |
| MERISE is a systemic method of analysis and design of information systems. That is, it uses a systems approach. | UML is however not a method but rather an object modeling language to which it is necessary to associate an approach to make it a method. This is the case with the 2TUP method; RUP and XP. |
| MERISE proposes to consider the real system from two points of view: - A static view (data) - A dynamic view (treatments). That is, with the MERISE method, we have a separate study of the data and the treatments. | UML offers a different approach from that of MERISE in that it combines data and processing. Because with UML, centralizing the data of a type and the associated processing makes it possible to limit the maintenance points in the code and facilitates access to information in the event of software development. In addition, UML describes the dynamics of the information system as a set of operations attached to the objects of the system. |
| Rational | Object |



B. COMPARATIVE STUDY OF UNIFIED PROCESSES

i. A Unified Process

A Unified Process is a process of development of software constructed on UML; it is iterative, incremental, centered on architecture, driven by use cases and requirements.

Iteration is distinct sequence of activities with a basic plan and evaluation criterion that produces an internal or external output. Either the content of an iteration is improved, or the evolution of the system is evaluated by users.

An increment is the difference between two released products at the end of two iterations. Each iteration that the group is capable of integrating the technical environment in order to develop a final product and give users the possibility of having tangible results.

Centered on architecture the different models derived during the establishment of system must be reliable and coherent.

Driven by use case and requirements enables the clear definition of a users' needs and priorities respectively thereby minimizing the risk of project failure.

ii. The Two Track Unified Process (2 TUP)

2TUP is a unified process which is built on UML and has as objective to bring solution to constraints of functional and technical changes imposed on information systems by strengthening controls on development capacities. It proposes a Y-shaped development life cycle that separates the functional aspect from the technical aspects, and the merging of these two forms the implementation aspect. 2TUP distinguishes therefore two branches: the functional and technical branches, the combination of the result of these two branches forms the third: the realization branch – where we realize our system. The diagram below illustrates the branches of 2TUP.

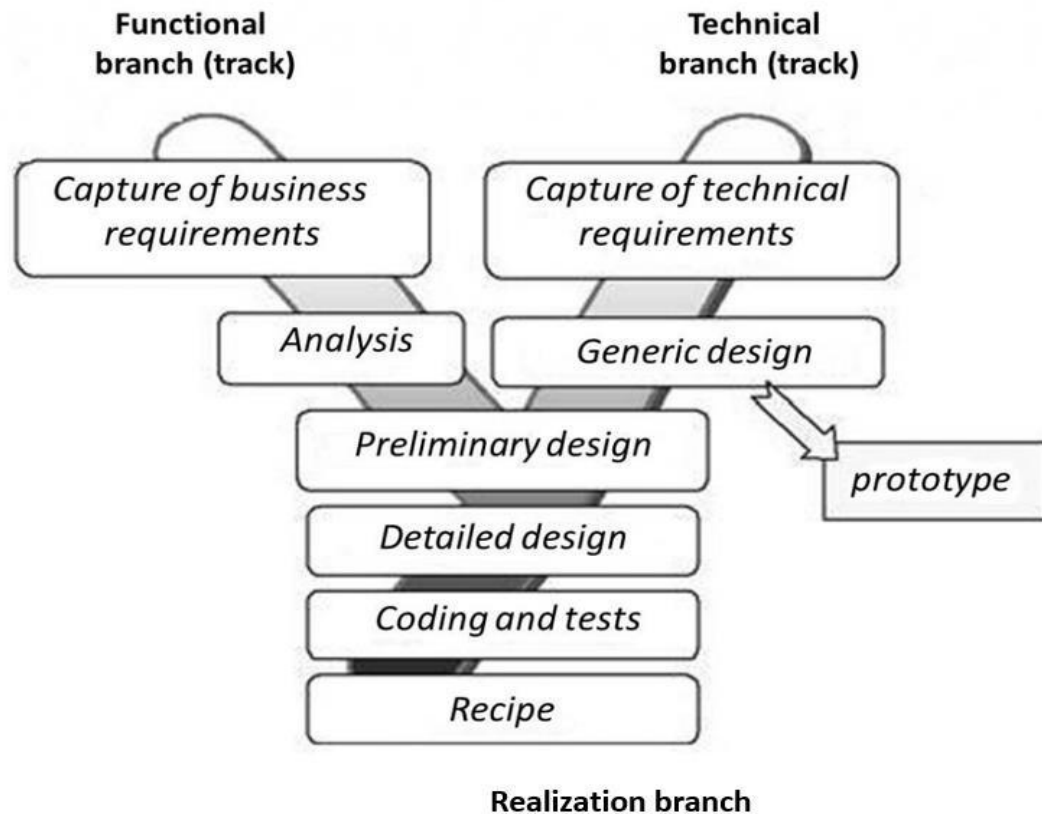


Figure 4: 2TUP diagram (source: 2TUP.COM)

a. The left branch (functional branch)

It captures the functional needs of a system. This ensures the production of software that meets the needs/requirements of the user. The analysis here consists of studying precisely the functional specification in order to obtain an idea of what the system is going to realize, and its result does not depend on any technology.

b. The right branch (Technical branch)

The technical branch enumerates the technical needs and proposes a generic design validated by a prototype. The technical needs include constraints and choices related to the conception of the system, the tools and equipment as well as the integration constraint with the existing system condition.



C. The middle branch (Realization or Implementation branch)

In this branch, we study the preliminary conception, detailed conception, and documentation of the system. The realization branch supports the following:

Preliminary conception: This is the most sensitive step of 2TUP as it is the confluence of the functional and technical branch. It is completed when the deployment model, the operating model, the logical model, interphases and the software configuration model are defined. We have the following diagrams:

- ❖ Component Diagram
- ❖ Deployment Diagram
- ❖ Package Diagram
- ❖ Composite Structure Diagram

Detailed conception: This is the detailed design of each feature of the system. We have the following diagrams:

- ❖ Class
- ❖ Object
- ❖ Sequence
- ❖ Timing Diagram

Coding and testing: This is the phase where we program the designed features and test the coded features.

The recipe: Also known as the deliverables is the validation phase of the functions of the developed system.



II. CHOICE OF THE ANALYSIS METHOD

The reason why we chose UML modelling language and the software development process 2TUP instead of many others that exist, includes:

1. UML is the current standard for programming in an object-oriented language. For this reason, it is widely understood and well known making it easy for a new programmer to join the project and be productive from the very first day.
2. UML diagrams allow teams to virtualize how a project is or will be working, and they can be used in any field, not just software engineering. The diagrams will allow teams to virtualize together how a system, or a process will work or did work. It can provide new ideas for how teams have to collaborate to achieve the goal of the workflow process.
3. 2TUP is centered around the creation and maintenance of a model, rather than the production of mountain documents.
4. 2TUP is user oriented as it permits the development of software that responds to the needs of the users through the study of the user needs.
5. 2TUP is iterative and incremental, hence it enables the project team to produce refined amelioration if necessary and easily integrate it in the already existing system.
6. 2TUP by permitting the project team identify and test the key functionalities of the system limits the risk related to building the system.



III. MODELLING OF THE PROPOSED SOLUTION

A. Capture of Functional Needs

The first step of the left (functional) branch of Two Track Unified Process (2TUP) is the capture of the functional needs. At this step, we capture the intended behavior of the system that maybe express as services, tasks or functions the system is required to perform.

i. USE CASE DIAGRAM

a. Definition

Use case diagram shows the functionalities of a system, their interdependencies and how they relate with actors of the system. A use case is a specification of behaviour. The main objectives of the use case diagram are:

- ❖ Provide a high-level view of the system.
- ❖ Identify the functions of the system.

Use case diagrams are completed with a textual description of each use case that is intended to define the use case in greater details.



b. Formalism

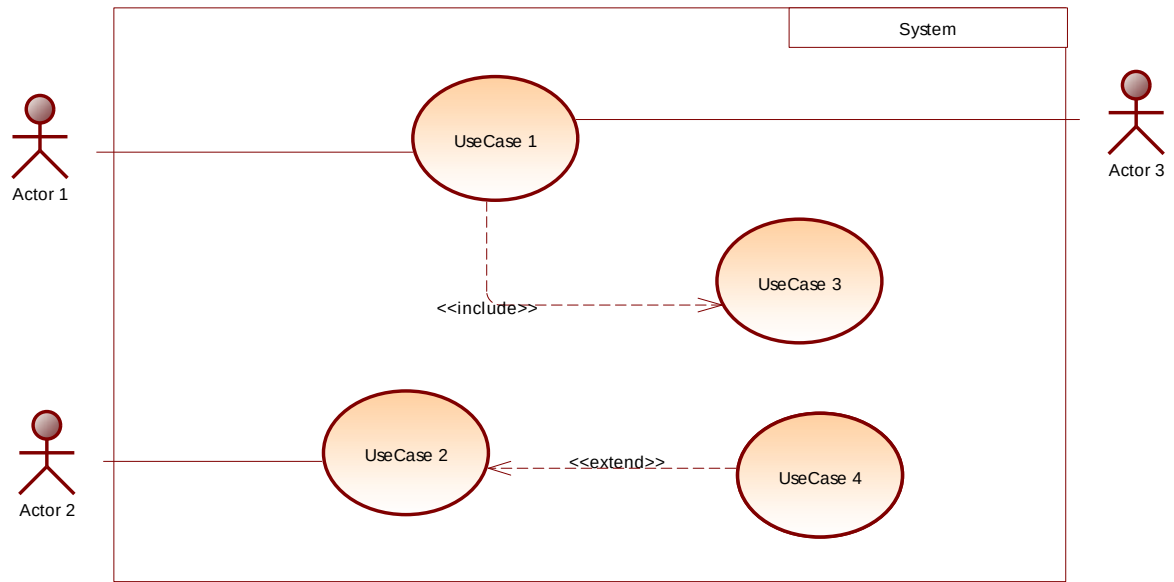
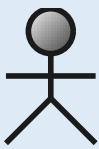
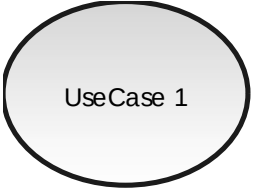



Figure 5: Use Case Diagram formalism

c. The Component of the use case Diagram

:

Table 9

| Elements | Notation | Description |
|-------------|--|---|
| Actors |  Actor 1 | Represents an entity that directly interacts with the system. The actor is what performs the different possible actions of the system |
| Use case |  UseCase 1 | A use case represents a functionality of the system. It is an action that can be performed by an actor. |
| Association |  Actor 1 UseCase 1 | it indicates that an actor takes part in a use Case. |



CONCEPTION AND IMPLEMENTATION OF AN AI-ASSISTED AGRIFUTURE INVESTMENT PLATFORM



| | | |
|---------|--|--|
| Include | | An inclusion denotes that an included action must be performed before the including action can be performed. |
| Extend | | An extension denotes that an extending action may be performed while an extended action is being performed. |

Table 10 :Use Case Diagram Component\

| | | |
|----------------|--|---|
| Generalization | | This shows that an actor or a use case is a kind of another abstract or concrete actors can be defined and later specialized using generalization relationship. |
| System | | It is a container of use cases which interact with external actors |



d. The Actors of our System

Table 11 Actors of our System

| Actor | Role |
|------------------------|---|
| Super Administrator | Manages the entire platform, including user accounts, project approvals, and system settings. |
| Farmer / Project Owner | Registers on the platform, submits agricultural projects for investment, and updates project information. |
| Contributors | Views available agricultural projects, makes funding decisions, and communicates with project owners. |
| Agri-Advisor | Provides expert advice to farmers, validates projects, and assists investors in making informed decisions. |
| Cloud Storage API | Used for securely storing documents, project media, and other system files (e.g., NextCloud or Firebase). |
| SMTP Server | Sends email notifications to users (e.g., registration confirmation, project approval, investment updates). |



CONCEPTION AND IMPLEMENTATION OF AN AI-ASSISTED AGRIFUTURE INVESTMENT PLATFORM



e. General Use Case Diagram



f. Consult document use case diagram

g. Consult workspace use case diagram



CONCEPTION AND IMPLEMENTATION OF AN AI-ASSISTED AGRIFUTURE INVESTMENT PLATFORM



h. Consult account list use case diagram



CONCEPTION AND IMPLEMENTATION OF AN AI-ASSISTED AGRIFUTURE INVESTMENT PLATFORM



i. Textual description of use cases



ii. COMMUNICATION DIAGRAM

a. Definition

Communication Diagrams model the interactions between objects in a sequence. They describe both the static structure and the dynamic behavior of a system. It is a simplified version of a Collaboration Diagram introduced in UML 2.0. A communication diagram is more focused on showing the collaboration of objects rather than the time sequence.

b. Formalism

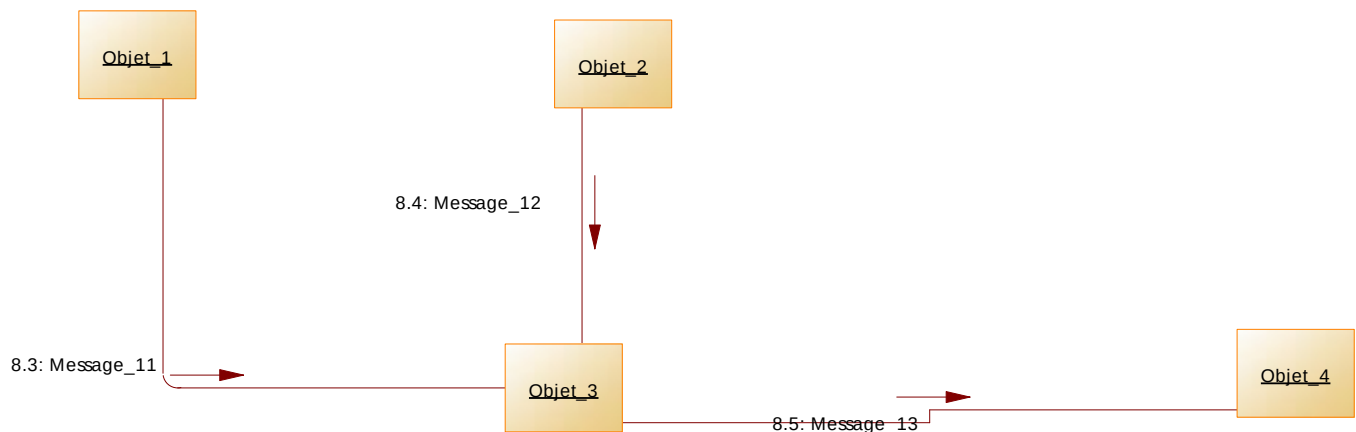




Figure 6: Communication Diagram formalism

c. Components of a Communication Diagram

Table 12 Elements of a communication diagram

| ELEMENT | NOTATION | DESCRIPTION |
|------------|-----------------|--|
| Message | 1: Message_1 → | Designs a particular communication between lifelines. |
| Connectors | 1: Message_1 —→ | It represents the relationships that exist between lifelines |



| | | |
|-------------------|--|---|
| Dependency |  | A dependency is a relationship that signifies a single or a set of model elements for their specification |
| Lifeline |  | An object represents an individual participant in the interaction conversation. |



iii. SEQUENCE DIAGRAM

a. Definition

A Sequence diagram describes interactions among classes in terms of an exchange of messages over time. They are also called event diagrams. A Sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modelling a new system.

b. Formalism

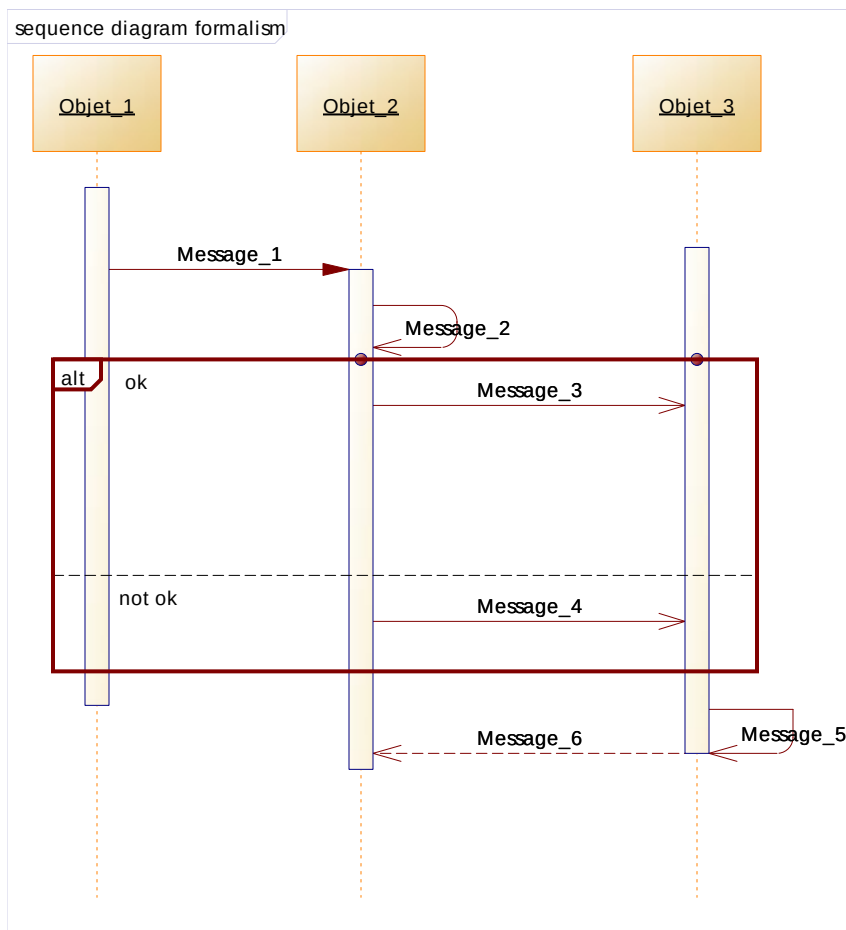

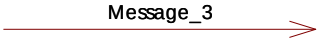
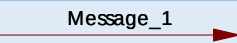
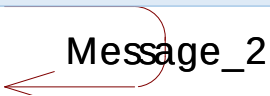

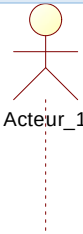
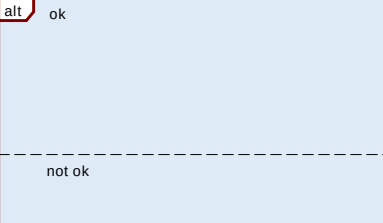


Figure 7 :Formalism of sequence diagram



C. Components of Sequence Diagram

Table 13: Elements of sequence diagram

| ELEMENT | NOTATION | DESCRIPTION |
|-------------------------------|---|---|
| Lifelines |  | They represent rows or objects instances that participate in the sequence being modelled. |
| Asynchronous Message |  | It is a message that receives an indirect response. |
| Synchronous Message |  | It is a message that sends and want response before it continues a process |
| Self-Message |  | self-message can represent a recursive call of an operation, or one method calling another method belonging to the same object. |
| Return Message |  | It represents the response of a message. |
| Actor |  | They send and receive message. |
| Combined Interaction Fragment |  | An articulation of interaction diagram, defined by an operator and operands. |



iv. ACTIVITY DIAGRAM

a. Definition

An activity diagram is a graphical representation of workflows that show the steps needed in the realization of a process; showing the details from a start point to an end point through all decisions and actions that can possibly be performed. Activity diagrams are intended to model both the computational and organizational process. They flow can be sequential, branched or concurrent. Below is an activity diagram formalism.

b. Formalism

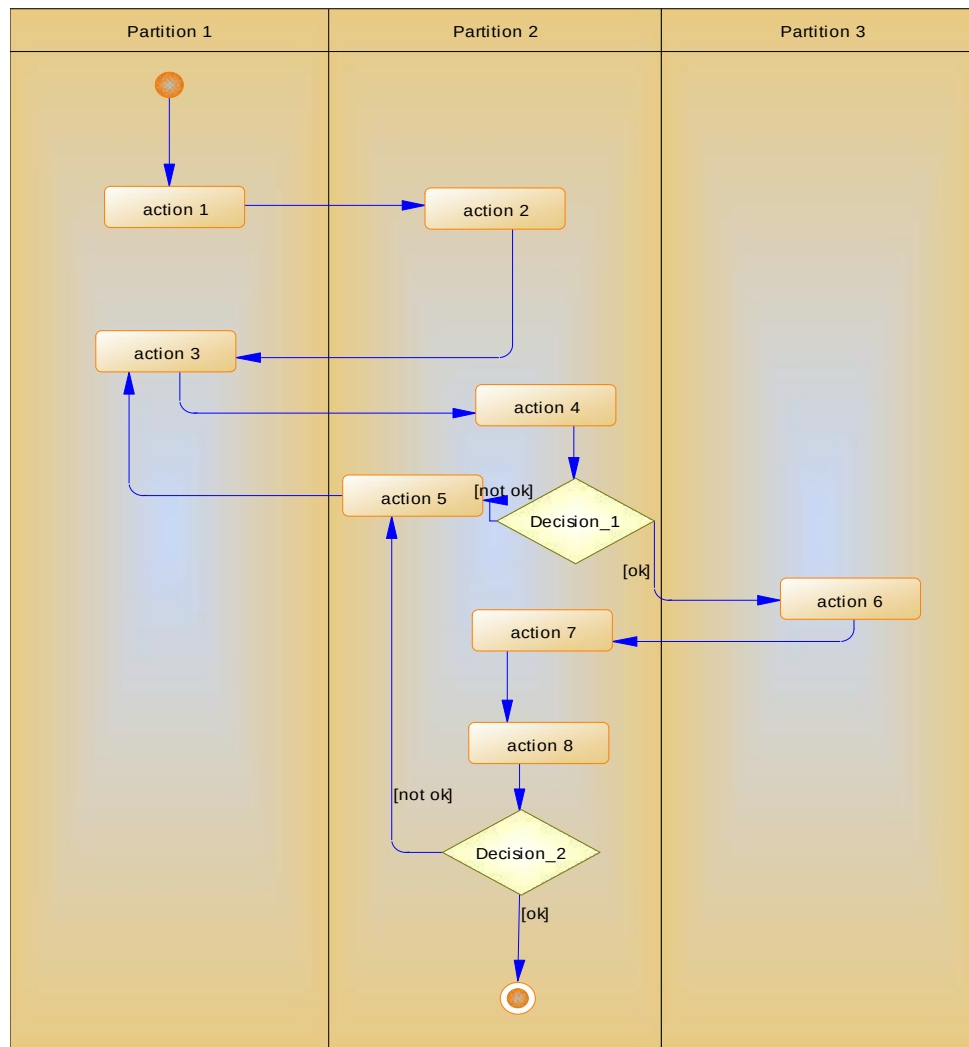

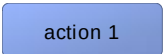



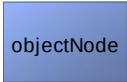
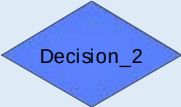
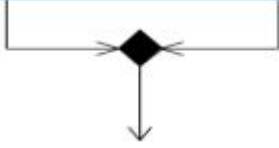


Figure 8 Formalism of activity diagram

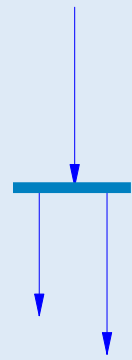
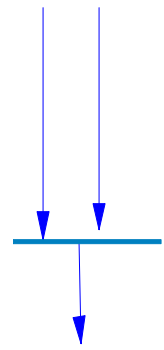
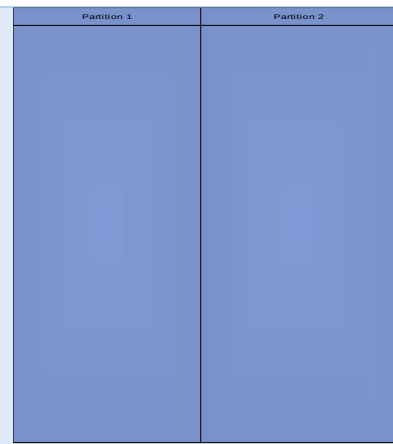


C. Elements of activity diagrams

Table 14 :The Component of the Activity Diagram

| Element | Diagrammatic Representation | Description |
|----------------------|---|--|
| Activity |  | Used to represent a set of actions. |
| Action |  | Represent a task to be performed. |
| Activity edge |  | A directed connection between two activity nodes through which tokens may flow |
| Initial node |  | Shows the beginning of an activity or set of actions. |
| Final node |  | Stops all controls and object flows in an activity. |
| Object node |  | Represents an object connected to a series of object flows. |
| Decision node |  | Represents a test condition that splits an incoming activity edge into opposite outgoing activity edges. |
| Merge node |  | Reunite different decision paths created using a decision node. |



| | | |
|-------------------------------|--|---|
| Fork node |  | Slits behaviour into parallel or concurrent flows of activities (or actions). |
| Join node |  | Unites a set of parallel or concurrent flows of activities or actions. |
| Swimlane and partition |  | A way of grouping activities performed by the same actor in an activity diagram or to group actions in the same thread. |



CONCLUSION

In the analysis phase, we selected a software development process and a modeling language, after which we outlined the functional requirements of our system. We examined the use case diagram, which illustrates the relationships between the actors and the use cases (the actions the actors can perform within the system). We also explored the communication diagram, which depicts the system's architecture based on object-oriented programming principles. Additionally, we analyzed the sequence diagram, which outlines the flow of messages between elements in the system. Finally, we reviewed the activity diagram, which illustrates the workflow of our system. We will now transition to the conception phase, where we will present the technical aspects of our system along with the associated diagrams.



PHASE IV: CONCEPTION PHASE



Preamble

The conception phase will permit us to present in an orderly manner the components necessary for the good functioning of our software and also the architecture used for the proposed solution. It bridges the gap between the analysis phase and the realization phase.

Content

INTRODUCTION

- I. TECHNICAL BRANCH
 - A. GENERIC DESIGN
 - B. CAPTURE OF TECHNICAL NEEDS
- II. RELATED UML DIAGRAMS
 - i. OBJECT DIAGRAM
 - ii. STATE MACHINE DIAGRAM
 - iii. PACKAGE DIAGRAM



INTRODUCTION

The conceptual phase will provide a detailed description of the necessary specifications, features, and operations required to meet the functional requirements of the proposed system, as outlined in the analysis phase. This stage aims to identify and consider essential components both hardware and software as well as the system's structure (including network capabilities), processes, and procedures needed to achieve its objectives. We will examine several key diagrams, including the object diagram, state machine diagram, and package diagram, to support our understanding of the system's design and functionality.



I. TECHNICAL BRANCH

A. GENERIC DESIGN

a) Hardware diagram of the system:

The hardware diagram simply shows how the system components of our system are deployed; it shows the positioning of each device into its right proportion.

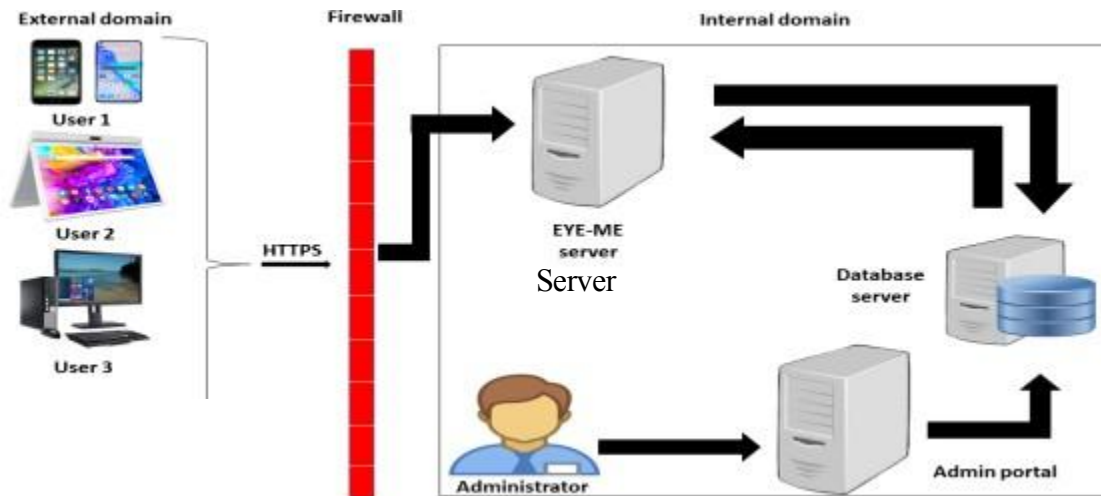


Figure 9 : Hardware Architectural Diagram of the system

b) High level architectural diagram of the software

The high-level architecture diagram provides an overview of the entire system, identifying the main components that would be developed for the product and their interfaces. The high-level architecture diagram below illustrates this.



B. CAPTURE OF TECHNICAL NEEDS

A. Physical Architecture

The design of the **AI-Assisted AgriFuture Investment Platform** is based on an **n-tier architecture**, which divides the system into independent yet interrelated layers. This modular approach facilitates scalability, maintainability, and ease of upgrading, as each tier can evolve independently without affecting others.

The **n-tier architecture** of the AgriFuture Investment Platform consists of the following components:

1. Client Tier (Presentation Interface)

This tier represents the **user interface** of the platform. It is accessed via a web application by all stakeholders including farmers, investors, agri-advisors, and administrators. It allows users to:

- Register and log in
- Submit or browse investment projects
- Book consultations
- Receive notifications

2. Application Tier (Business Logic Layer)

This tier contains the **core business logic**. It manages user requests, handles authentication and authorization, performs validations, processes investment-related workflows, and acts as the link between the user interface and data layer. It is deployed on a web server.

3. Data Tier (Database Management Layer)

This tier manages **all persistent data** related to:

- User profiles (farmers, investors, advisors)
- Project submissions and funding records
- Notifications and messages
- System logs

Data is stored using a **relational database** such as MySQL or PostgreSQL, with emphasis on data integrity, security, and performance.

4. AI/LLM Tier (AI-Assisted Intelligence Layer)

This additional tier integrates **Large Language Model (LLM)** functionality to provide intelligent features such as:

- Project proposal analysis and feedback for farmers
- Investment recommendation support for investors



- Automated summarization of projects
- Natural language interaction (chatbot or smart assistant)

This tier operates through an AI server (e.g., an instance of Ollama or another LLM deployment).

5. Email Tier (Notification and Communication Layer)

This tier includes the **SMTP mail server**, which is responsible for sending email notifications such as:

- Account registration confirmations
- Project approval or rejection alerts
- Investment status updates
- General system alerts

The email server integrates with Gmail or another SMTP service.

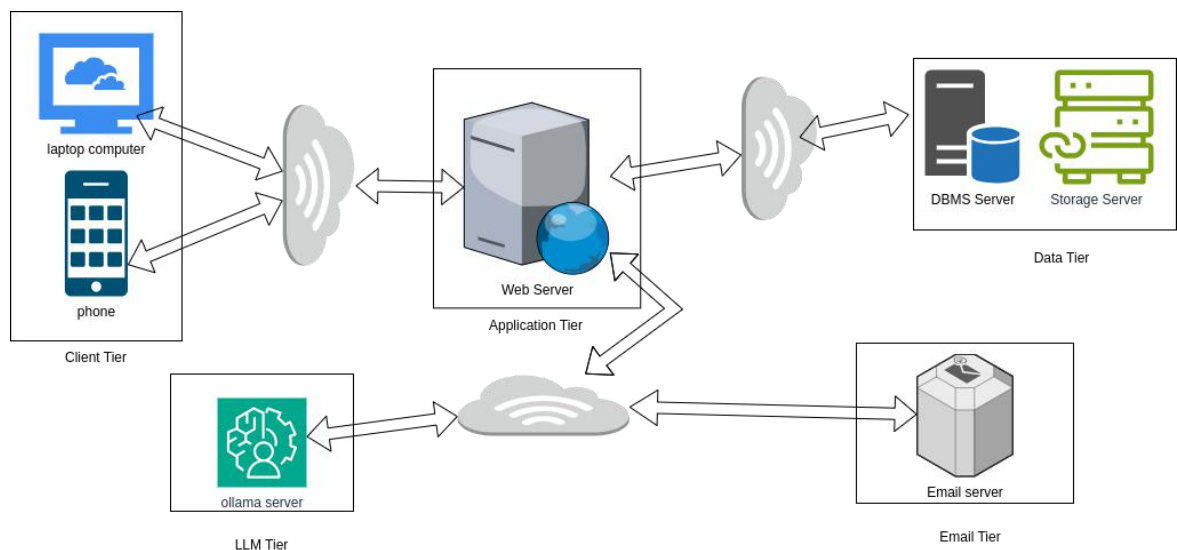


Figure 10



B. Logical Architecture

Model View controller or MVC as it is popularly called, is a software design pattern for developing application. A model view controller pattern is made up of the following three parts.

- ❖ **Model:** The lowest level of the pattern which is responsible for maintaining data.
- ❖ **View:** This is responsible for displaying all or a portion of data to the user.
- ❖ **Controller:** It handles software codes that controls the interactions between the model and the view.

MVC is popular as it isolates the application logic from the user interface and supports separation of concerns. Here the controller receives all requests for the application then works with the model to prepare data needed by the view. The view then uses the data prepared by the controller to produce a final response. The MVC can be represented as follows:

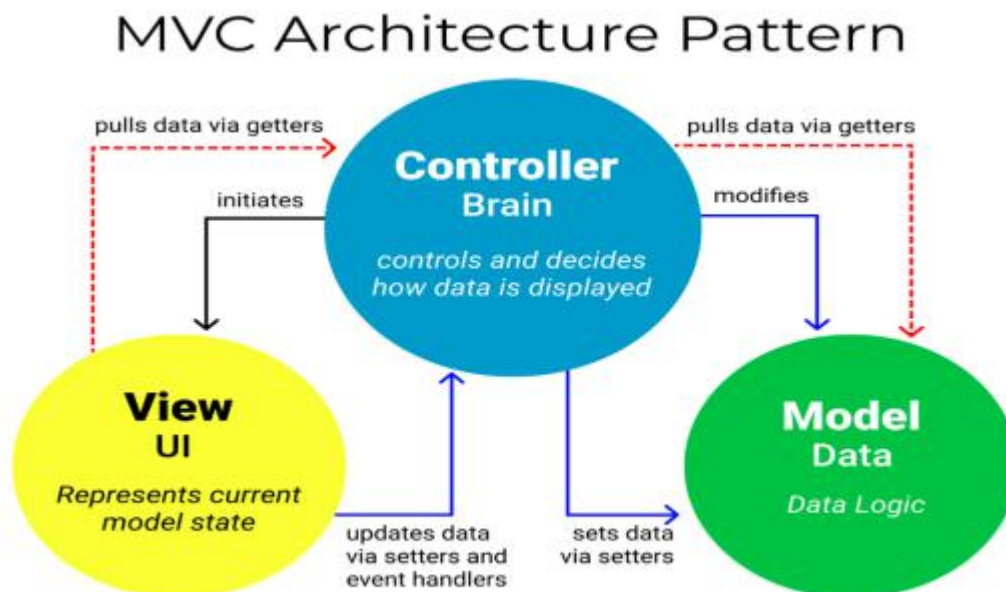


Figure 11 The MVC architecture (Source: <https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained/>)



II. RELATED UML DIAGRAMS

i. CLASS DIAGRAM

a. Definition

A class diagram is a static diagram. It represents the static view of an application. class diagram is not only used for visualizing, describing and documenting different aspect of the system but also for constructing executable code of the software application. Class diagram describes the attribute and operation of a class and constraints imposed on the system. Its purpose is to model the static view of an application.

b. Formalism

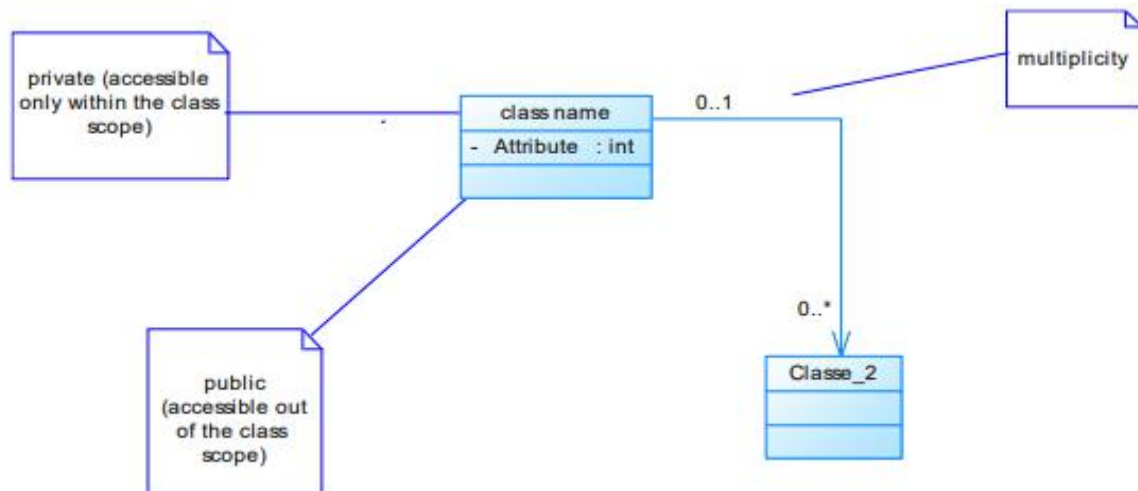


Figure 12 : Formalism of Class Diagram

c. Elements of the Class Diagram

Table 15 : Element of the Class Diagram

| Element | Representation | Description |
|-------------|----------------|---|
| Class | | A class is an element that defines the attributes and behaviors that an object can generate |
| Composition | | If a parent of a composite is deleted, usually, all its parts are deleted with it. |

Written by : Somo Rean Giggis Kinyuy,
a 2nd year student studying Software Engineering at AICS Cameroon, Paul Biya Technological Center of Excellence, for the Academic year 2024–2025



CONCEPTION AND IMPLEMENTATION OF AN AI-ASSISTED AGRIFUTURE INVESTMENT PLATFORM



| | | |
|-----------------------|--|---|
| Aggregation | | If the parent of the aggregate is deleted, usually the children are not deleted. |
| Dependency | | It existed between two classes, if one changes it may cause the change in the order, but the other way around |
| Generalization | | it a relationship between a whole thing (called superclass) and a more specific thing (called subclass) |
| Association | | It is a general type of relationship between elements, it may include cardinality, roles etc. |



d. Agrifuture class diagram



ii. OBJECT DIAGRAM

a. Definition

Object is an instance of a class in a particular moment in runtime that can have its own state and data values. Likewise, a static UML object diagram is an instance of a class diagram, it shows a snapshot of the detailed state of a system at a point in time, thus an object diagram encompasses objects and their relationships which may be considered a special case of a class diagram or a communication diagram.

b. Formalism

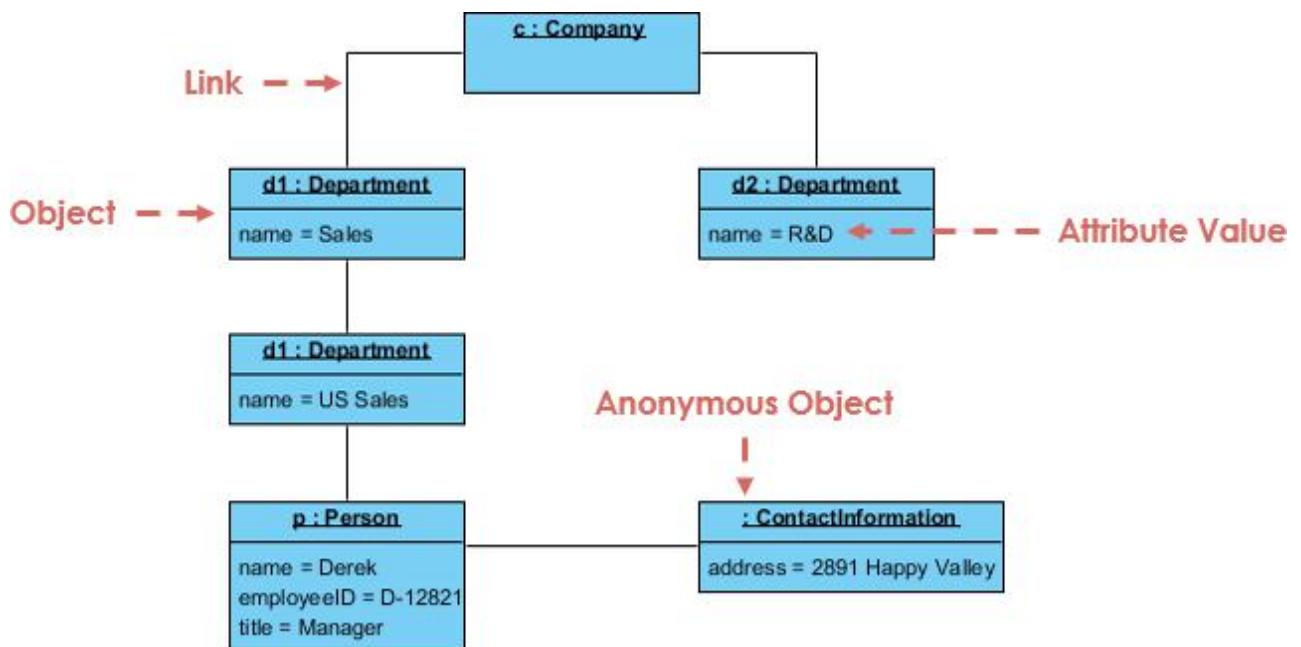
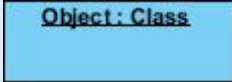
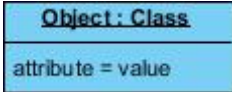
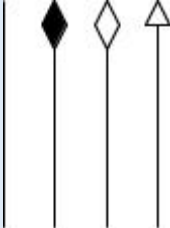


Figure 13 : Formalism of Object Diagram



C. Elements of Object Diagram

Table 16 Elements of object diagram

| Element | Representation | Description |
|-------------------|---|--|
| Object Names |  | Every object is actually symbolized like a rectangle, that offers the name from the object and its class underlined as well as divided with a colon. |
| Object Attributes |  | Similar to classes, you are able to list object attributes inside a separate compartment. However, unlike classes, object attributes should have values assigned for them. |
| Links |  | Links tend to be instances associated with associations. You can draw a link while using the lines utilized in class diagrams. |



d. Agrifuture object diagram



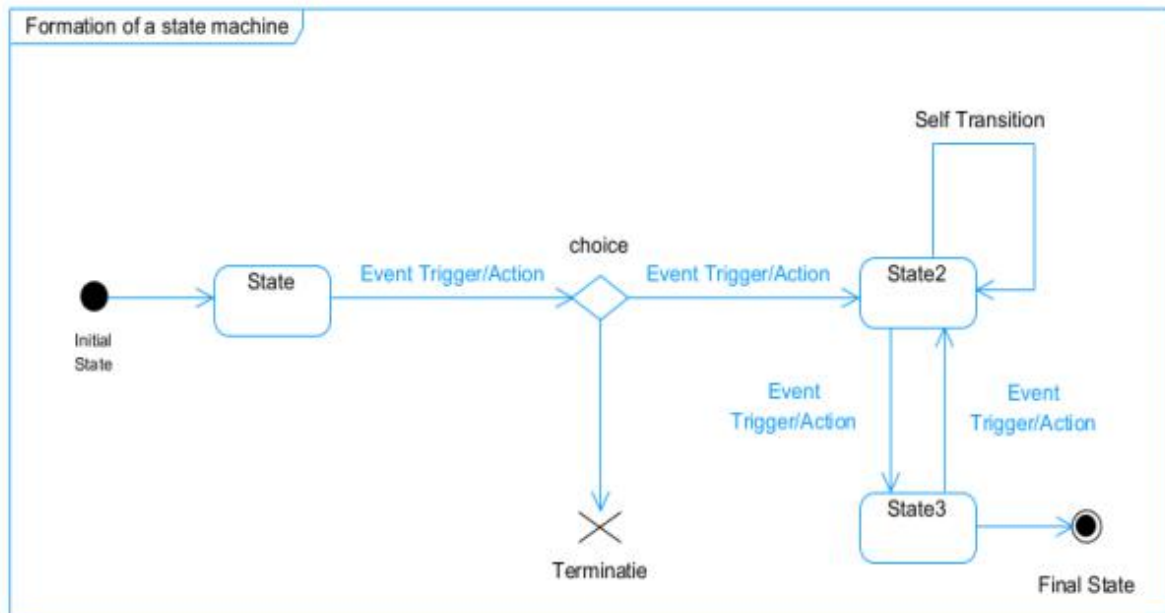
iii. STATE MACHINE DIAGRAM

a. Definition

A state machine diagram describes the behaviour of a single object in response to a series of events in a system. Also known as the state machine diagram, it models the dynamic flow of control from the state of a particular object within a system.

b. Formalism

Figure 14 : Formalism of state machine diagram



Written by : Somo Rean Giggis Kinyuy,

a 2nd year student studying Software Engineering at AICS Cameroon, Paul Biya Technological Center of Excellence, for the Academic year 2024–2025



C. Elements of State Machine Diagram

Table 17 : Elements of state machine diagram

| Element | Representation | Description |
|-----------------------|----------------|---|
| State | | Models a situation during which a certain invariant condition holds. |
| First (Initial State) | | It represents a default vertex, that is, a source for a single transaction to the default or composite state. |
| Final State | | A state specifying that the enclosing region is complete. |
| Transition | | A direction relation between a source and a target vertex. |
| Choice pseudo state | | A diamond symbol that indicates a dynamic condition with branched potential results |
| Terminate | | Implies that the execution of a state by means of its context is terminated. |
| Diagram overview | | A placeholder for the linked states in a state machine diagram. |

Written by : Somo Rean Giggis Kinyuy,

a 2nd year student studying Software Engineering at AICS Cameroon, Paul Biya Technological Center of Excellence, for the Academic year 2024–2025



CONCEPTION AND IMPLEMENTATION OF AN AI-ASSISTED AGRIFUTURE INVESTMENT PLATFORM



Diagrams here



iv. PACKAGE DIAGRAM

a. Definition

This is a structural diagram used to show the organization and arrangement of various model elements in the form of packages. A package diagram is the grouping of related uml elements such as classes, diagrams or even other packages.

b. Formalism

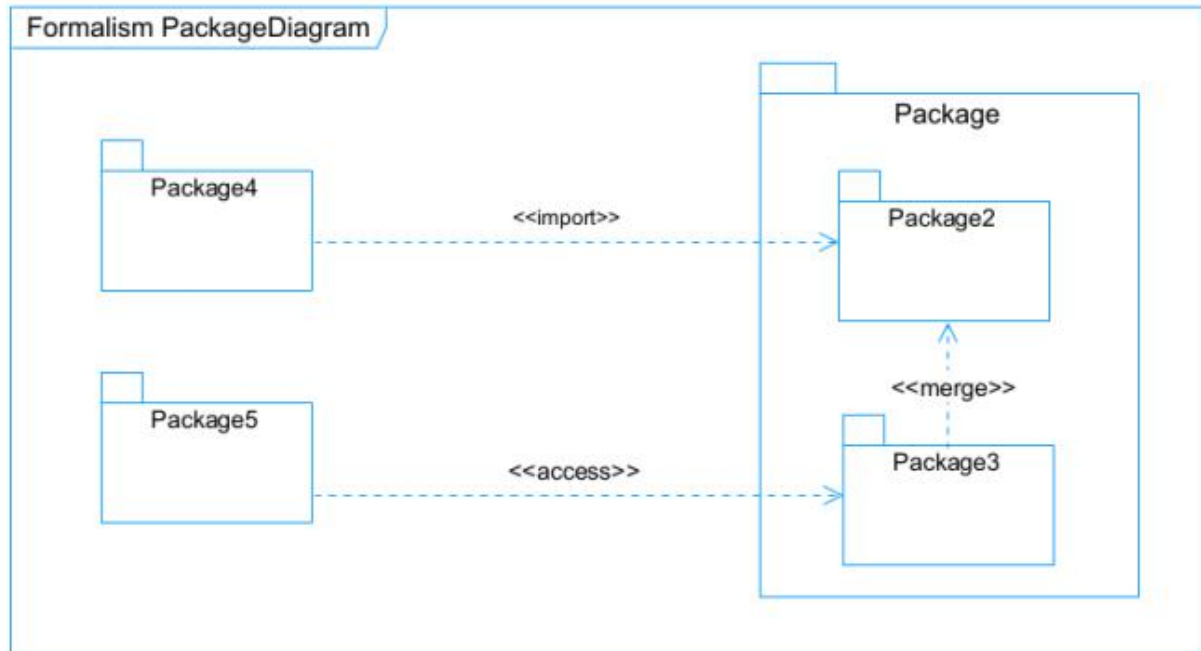
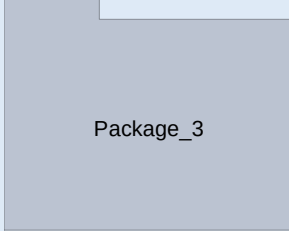
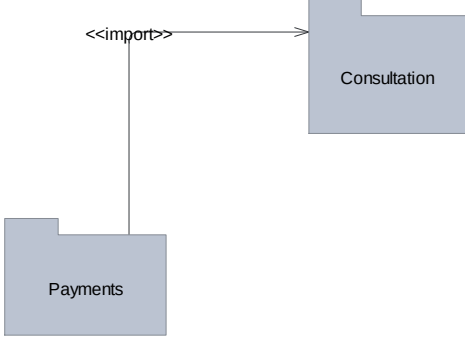
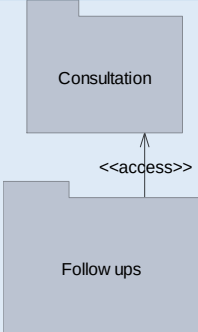
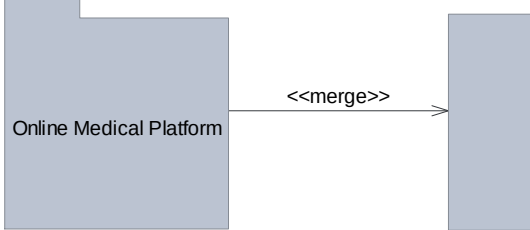


Figure 15: Formalism of package diagram



C. Elements of the package Diagram

Table 18: Elements of package diagram

| Element | Representation | Description |
|-----------------------|--|---|
| Package |  | A package is a namespace use to group related elements; it is a mechanism used to group elements into a better structure in a system. |
| Package import |  | A relationship Indicate that, functionality has been imported from one package to another. |
| Package access |  | A relationship Indicates that one package requires assistance from the function of another package. |
| Package merge |  | It is a relationship which shows that, the functionality of two packages are combines to a single function. |



CONCEPTION AND IMPLEMENTATION OF AN AI-ASSISTED AGRIFUTURE INVESTMENT PLATFORM



d. Agrfuture package diagram



Conclusion

In the **conception phase**, our primary objective was to outline and structure the core components of the **AgriFuture Investment Platform**, ensuring it fits seamlessly within a modern technical architecture. We began by identifying the technical constraints and system requirements necessary for the platform's successful deployment.

We then proceeded to define the physical and logical architecture of the system, specifying the roles of each component and how they interact with one another. Additionally, we highlighted the key system actors and their respective responsibilities, ensuring a clear understanding of user interaction and system behavior.

This phase has laid the groundwork for implementation by providing a solid architectural foundation. The next step of our project is the **realization (implementation) phase**, where we will bring the design to life through actual development and system integration.



PHASE V: REALIZATION PHASE



Preamble

In this phase we will to straight forward in the implementation of our solution, we will base ourselves on the analysis and conception phases and also present the component and deployment diagrams.

Content

INTRODUCTION

I. TECHNOLOGY STACK

II. RELATED UML DIAGRAMS

A. DEPLOYMENT DIAGRAM

B. COMPONENT DIAGRAM

CONCLUSION



Introduction

The **realization phase** marks the critical transition from system design to actual implementation of our **AI-Assisted AgriFuture Investment Platform**. This phase focuses on the application of the previously defined architectural design, using a selected technology stack and supporting UML diagrams—specifically, the **deployment** and **component diagrams**.

We begin by presenting the technologies chosen for the **frontend**, **backend**, **database**, and **cloud services**, ensuring the platform is both scalable and performance-optimized. The **deployment diagram** will provide a visual representation of how the application components are distributed across various servers and services in the physical infrastructure. Additionally, the **component diagram** will outline the internal structure of the system, showcasing the interactions between major modules such as user management, project submission, AI-assisted analysis, and notification services.

This comprehensive and systematic approach ensures a strong foundation for development, promotes collaboration among team members, and maintains alignment with the project's objectives—ultimately facilitating a successful and impactful deployment of the AgriFuture Investment Platform.



I. Technology Stack

The development of the **AI-Assisted AgriFuture Investment Platform** relies on modern and reliable technologies to ensure scalability, maintainability, and high performance. The table below presents the key technologies used throughout the development process:


Table 19: Used Technology in Development Process

| Technology | Logo | Description | Reason for Use |
|----------------------|------|---|---|
| React.js | | A popular JavaScript library for building user interfaces. | Provides a fast, responsive, and modular frontend with reusable components. |
| Node.js | | A JavaScript runtime for building fast and scalable backend applications. | Enables real-time interactions and seamless integration with frontend via APIs. |
| Express.js | | A minimalist web framework for Node.js. | Facilitates API development and backend routing. |
| MongoDB | | A NoSQL document-oriented database. | Allows flexible storage of user and project data with easy scalability. |
| Firebase | | A platform by Google offering real-time database and cloud storage. | Used for authentication, notifications, and media storage. |
| Ollama / LLM Server | | AI engine integrating LLM capabilities. | Powers the AI-assisted project review and investment recommendation features. |
| NextCloud (Optional) | | Cloud-based file storage and collaboration service. | Used for securely storing and managing documents and |



CONCEPTION AND IMPLEMENTATION OF AN AI-ASSISTED AGRIFUTURE INVESTMENT PLATFORM



| | | | |
|--------------------|---|---|---|
| | | | multimedia assets. |
| Gmail SMTP | | Google's email server protocol. | Enables sending automated emails (e.g., registration, updates, approvals). |
| Git & GitHub | | Version control system and code hosting platform. | Ensures collaborative development and version tracking. |
| Visual Studio Code |  | A powerful and lightweight code editor. | Provides rich development features like debugging, Git integration, and extensions. |



I. Related UML DIAGRAMS

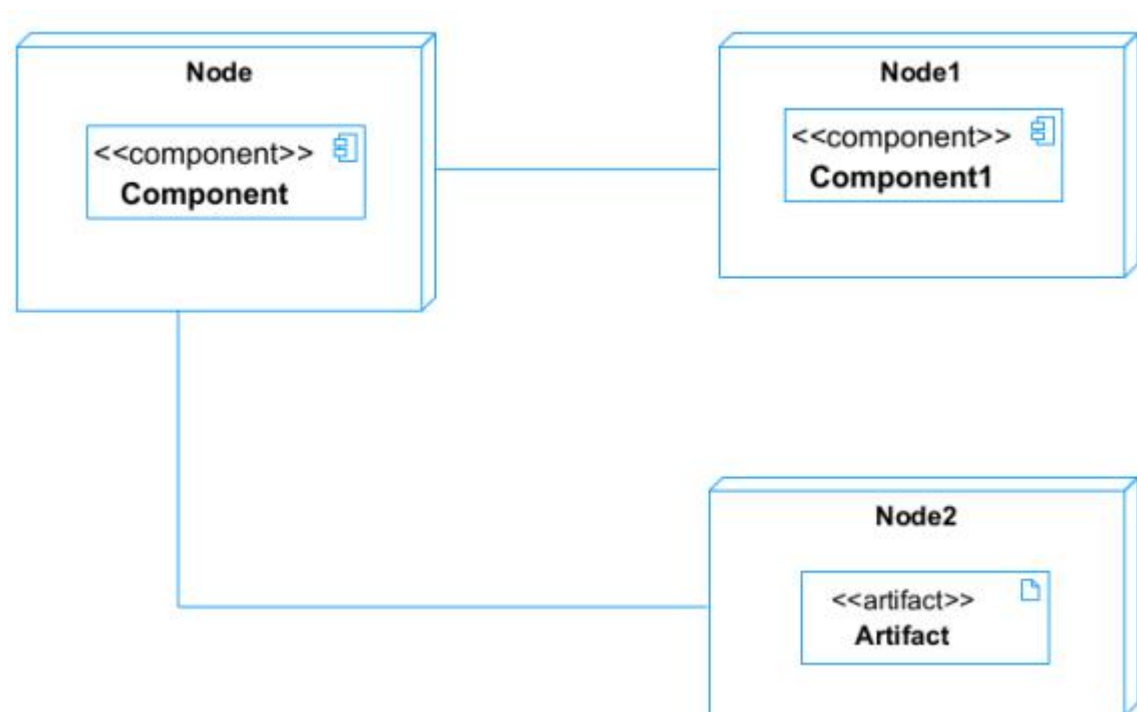
A. DEPLOYMENT DIAGRAM

a. Definition

Deployment diagram is a structural diagram used to visualize the topology of the physical components of a system, where the software is deployed. They consist of nodes and their relationship. It is related to the component diagram because the components are deployed using the deployment diagram. A deployment diagram consists of nodes. Nodes are nothing but physical hardware used to deploy the application.

b. Formalism

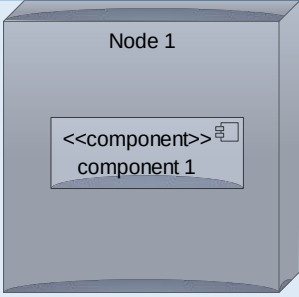

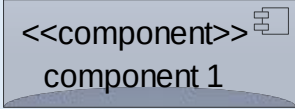
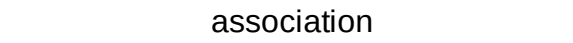
Figure 16 : Formalism of deployment diagram





C. Element of deployment diagram

Table 20: Elements of deployment diagram

| Element | Representation | Description |
|-------------|--|--|
| Node |  | It is a hardware used to deploy the application |
| Artifact |  | An artifact is a major product, which is produced or used during the development of a software. E.g diagrams, data models, setup scripts |
| Component |  | It represents a modular part of a system that encapsulates its content and whose manifestation is replaceable within its environment. |
| Association |  | An association helps to connect two nodes together which permits them to communicate together |



d. System deployment diagram



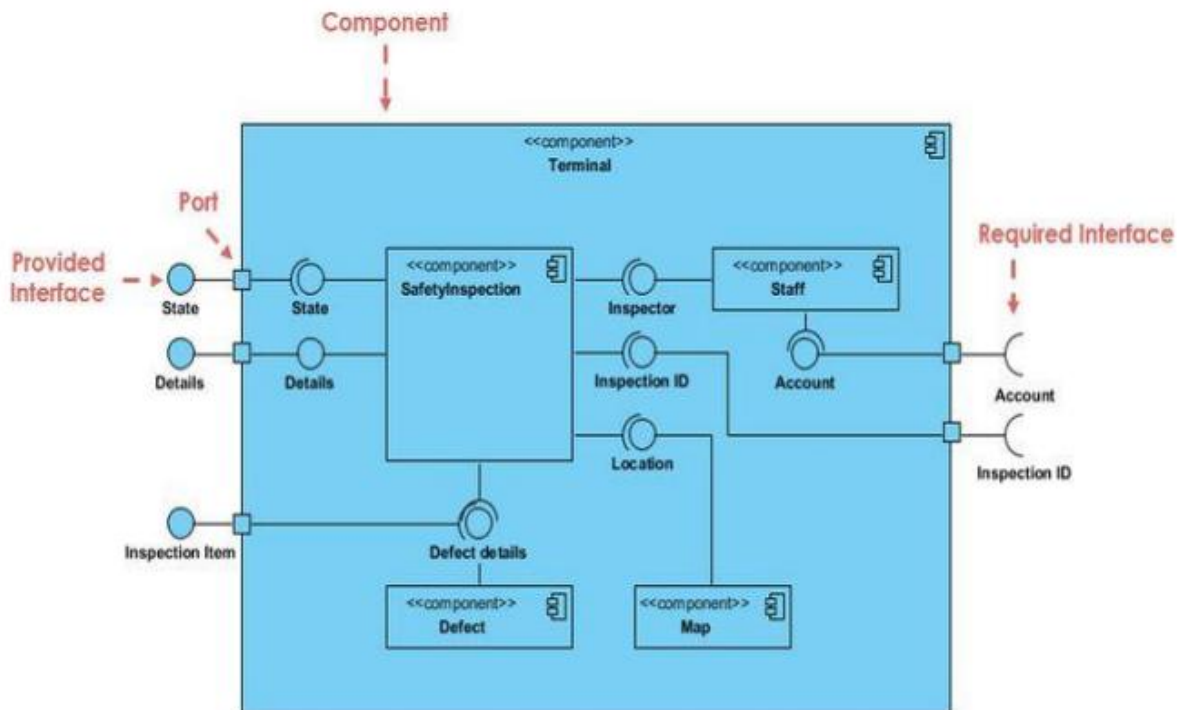
B. COMPONENT DIAGRAM

a. Definition

Component diagrams are used to model the physical aspect of a system. Now the question is what are this physical aspect? They are elements such as Executables, libraries, files, document etc. which resides in a node. The component diagram does not describe the functionality of the system, but it describes the components used to make those functionalities.

b. Formalism

Figure 17 : Formalism of component diagram (Source: <https://www.pinterest.com/pin/551128073157994549/>)



Written by : Somo Rean Giggis Kinyuy,

a 2nd year student studying Software Engineering at AICS Cameroon, Paul Biya Technological Center of Excellence, for the Academic year 2024–2025



C. Elements of Component Diagram

Table 21: Elements of component diagram

| Name | Representation | Description |
|---------------------------|----------------|---|
| A Component | | A component is an abstract logical unit block of a system.it is represented as a rectangle with smaller rectangle in the upper right corner which saves as it icon for recognition. |
| Dependency | | Dependency is a directed relationship which is used to show that some components are dependent on others for their correct functioning. |
| Required Interface | | It is a straight line from the component box with an attached half circle representing interfaces where a component requires information in order to perform its own functions. |
| Provided Interface | | It is a straight line from the component box with an attached circle representing interfaces where a component produces information used by required interfaces |
| Port | | A port (represented by a small square at the end of a required or provided interface) is used when the components delegate |

Written by : Somo Rean Giggis Kinyuy,

a 2nd year student studying Software Engineering at AICS Cameroon, Paul Biya Technological Center of Excellence, for the Academic year 2024–2025



CONCEPTION AND IMPLEMENTATION OF AN AI-ASSISTED AGRIFUTURE INVESTMENT PLATFORM



the interfaces to an internal class.

Written by : Somo Rean Giggis Kinyuy,

a 2nd year student studying Software Engineering at AICS Cameroon, Paul Biya Technological Center of Excellence, for the **Academic year 2024–2025**



d. System component diagram



Conclusion

In conclusion, the **realization phase** represents a vital milestone in the development of the **AI-Assisted AgriFuture Investment Platform**, where we translate our architectural designs into a working system through the implementation of the selected technology stack and the modeling of key UML diagrams.

By utilizing modern technologies such as **React.js**, **Node.js**, **MongoDB**, **Firebase**, and optionally integrating **AI/LLM services**, we ensure the platform is **scalable**, **efficient**, and tailored to meet the real needs of both farmers and investors. The **deployment diagram** illustrates how components are distributed across the infrastructure, while the **component diagram** clarifies the interactions and structure within the system. Together, these tools foster clearer team communication and a more organized development workflow.

With the implementation structure now in place, we are prepared to move forward into the **functional testing phase**, where the system will be thoroughly evaluated to ensure all features operate as intended. This will strengthen the reliability, security, and user experience of the platform before its final deployment.



PHASE VI: TEST OF FUNCTIONALITIES



Preamble

In this phase, we will present the various functionalities of our application.

Content

INTRODUCTION

I. APPLICATION FUNCTIONALITIES

II. TESTS SHOWCASES

CONCLUSION



INTRODUCTION

The testing phase allows us to gain deeper insights into the solution we are developing, for web platforms. This phase highlights the various functionalities or modules within our application and their benefits to users. In this context, we will explore the different functionalities present in **Agrifuture Investment PLatform**, examining how each module contributes to enhancing user experience and meeting their needs.



a) APPLICATION FUNCTIONALITIES

1. User Authentication and Role Management

- Secure login for **farmers, investors, firms, and admins.**
- Role-based dashboard and permissions.
- Admin controls for activating, suspending, or resetting user accounts.

2. AI Investment Advisor Chatbot

Interactive AI assistant that:

- Recommends viable agricultural projects for investment.
- Detects potential fraudulent activities based on user/project history and patterns.
- Generates professional project proposals for farmers and firms.
- Answers investor and farmer queries in natural language.

3. Project Submission and Evaluation

- Farmers and firms can submit detailed project proposals.
- AI-assisted analysis to predict project success likelihood and risk.
- Investors can evaluate, bookmark, and comment on projects.

4. Investment Matching Engine

- Matches investors with high-potential agricultural projects.
- Uses AI to analyze market trends, regional potential, and investment history.

5. Due Diligence and Fraud Detection

- AI-driven background checks on projects, users, and firms.
- Red flags suspicious behavior or duplicate/fake submissions.
- Notifies admin and investor of inconsistencies.

6. Proposal Generation & Optimization

- Farmers can use AI to auto-generate or refine funding proposals.
- Ensures standardized, professional, and convincing document structures.

7. Marketplace for Agricultural Solutions

- Firms can list services, tools, or machinery.
- Farmers and investors can browse offerings relevant to ongoing or proposed projects.

8. Collaboration & Messaging

Real-time chat between:

Written by : **Somo Rean Giggis Kinyuy**,
a 2nd year student studying Software Engineering at AICS Cameroon, Paul
Biya Technological Center of Excellence, for the **Academic year 2024–2025**



- Farmers and investors
- Firms and project stakeholders
- Internal team and AI assistant
- Supports discussion threads on specific projects.

9. Project Tracking & Milestone Reporting

- Once funded, projects have a dashboard to report milestones.
- Investors can monitor progress and receive updates.
- AI assesses deviations and recommends corrective action.

10. Analytics Dashboard

Insights for each user type:

- **Farmers:** Funding history, project performance
- **Investors:** Portfolio stats, ROI projections
- **Firms:** Demand metrics and engagement
- AI-driven recommendations to improve outcomes.

11. Notifications and Alerts

- Automated updates for project milestones, funding, chat messages, fraud detection, etc.
- Push/email notifications based on user preferences.

12. Task Management

- Assign tasks (e.g., submit report, update documents) within projects.
- Track task statuses: "Created", "In Progress", "Done", "Suspended", etc.

13. Knowledge Base and Help Center

- Access to FAQs, tutorials, best practices, and video walkthroughs.
- AI chatbot can guide users through the app and answer contextual questions.

14. Data Backup and Security

- All user data and project info backed up in secure cloud storage.
- Recovery tools for lost data, encrypted storage for sensitive details.

15. Ratings & Reviews

- Investors can rate and review completed projects.
- Firms and farmers build credibility over time through verified performance.



b) TESTS SHOWCASES



CONCLUSION

In the testing phase, we outlined and validated the various functionalities of the **AI-Assisted AgriFuture Investment Platform**, highlighting how the application benefits its key user groups including **farmers, investors, firms, and super administrators**. This phase was essential in ensuring that all features such as AI-driven project recommendations, fraud detection, intelligent proposal generation, and real-time collaboration functioned effectively and aligned with user needs.

The results confirmed that the platform enhances transparency, facilitates smarter agricultural investments, and improves communication between stakeholders. We are now ready to proceed to the final phase: the **Installation and User Guide**, where we will provide step-by-step instructions on how to set up and use the platform efficiently.



PHASE VII: INSTALLATION GUIDE AND USER GUIDE



Preamble

The purpose of the user guide is to provide users of our platform with step-by-step instructions on how to install and use the system.

Content

INTRODUCTION

- I. INSTALLATION OF THE APPLICATION
- II. SHOW CASE

CONCLUSION



INTRODUCTION

This is the final phase of our report. In this section, we will outline the requirements for our system, detail the installation process, and explain how to access the system and its features. We will present this information in a step-by-step manner to ensure that first-time users can easily set up the platform. Each step in the various processes will be accompanied by images for clarity. Following this, we will showcase our application by reviewing the different screens, highlighting the user interface and key functionalities.



I. INSTALLATION OF THE APPLICATION



I. SHOWCASE



CONCLUSION

Once the platform is set up, it is essential to provide a manual to assist its various users. Therefore, we have outlined the different tools required for installation and detailed the installation process to ensure the application runs smoothly. Additionally, we will explain how users can effectively navigate and utilize the platform once the environment is fully configured. This manual will serve as a comprehensive guide to maximize the user experience.



PERSPECTIVES



GENERAL CONCLUSION

Having reached the conclusion of our personal project, we have achieved significant milestones throughout this journey. The process has been both challenging and rewarding, pushing us to overcome obstacles while also offering valuable opportunities for learning and growth. This experience strengthened our ability to work in a professional environment, collaborate as a team, and remain solution-oriented in the face of setbacks.

Our project was centered on the theme **“Design and Implementation of an AI-Assisted AgriFuture Investment Platform,”** which aims to bridge the gap between farmers, investors, and agricultural firms using modern technologies and artificial intelligence. The primary goal was to create an intelligent ecosystem that simplifies agricultural investment by offering project recommendations, detecting fraudulent activities, and generating high-quality proposals for investment consideration.

We started by identifying the core needs of each user role—farmers seeking funding, investors exploring viable opportunities, and firms offering agricultural services. Through careful system analysis and iterative design, we defined clear functionalities using the UML-2TUP methodology to build a robust, scalable platform. The technologies employed include React for the user interface, Node.js and Express for the backend logic, MongoDB for data management, and machine learning libraries for AI-powered functionalities.

The platform offers a wide range of features, including intelligent investment matching, fraud detection, AI-generated proposals, chatbot assistance, and real-time messaging—each contributing to a streamlined and secure user experience. The integration of AI not only automates critical decisions but also enhances trust, efficiency, and transparency within the agricultural investment process.

Looking forward, we see numerous opportunities to evolve and refine our platform. Future enhancements may include deeper financial analytics, predictive project success modeling, integration with mobile payment gateways, and expanded multilingual support to reach broader rural communities. Our vision is to transform agricultural investing into a data-driven, intelligent process that empowers all stakeholders and drives sustainable growth in the sector.

This is only the beginning. We are committed to continuous improvements in functionality, user experience, and system security, ensuring that our platform remains a dependable and innovative tool that meets the evolving needs of modern agriculture and investment.



ANNEXES



BIBLIOGRAPHY

- ❖ Rowley, J. (2014) Writing Reports: A Guide for Students and Professionals. 2nd edn. Oxford: Oxford University Press.
- ❖ Day, R.A. and Gastel, B. (2012) How to Write and Publish a Scientific Paper. 7th edn. Cambridge: Cambridge University Press.
- ❖ Oshima, A. and Hogue, A. (2006) Writing Academic English. 3rd edn. New York: Pearson Longman.
- ❖ Murray, N. and Nicol, D. (2014) Promoting Good Academic Writing in Higher Education. London: Routledge.



WEBOGRAPHY

- ❖ https://www.tutorialspoint.com/object_oriented_analysis_design/index.htm
Object Oriented Analysis & Design Tutorial. August 15th, 2024, 09:00 PM.
- ❖ <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/> What is a Class Diagram? April 10th, 2025, 09:00 PM.
- ❖ <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-object-diagram/> What is an Object Diagram? April 10th, 2025, 11:00 PM.



TABLE OF CONTENT