# **DEDICATION**

THIS WORK IS DEDICATED TO MY FAMILY

# ACKNOWLEDGEMENT

Drafting this document would have not been possible without the contribution of some people who took upon themselves to see this work being accomplished. Our gratitude goes to the following people:

* The Resident Representative of IAI-Cameroon, **Mr. ARMAND CLAUDE ABANDA**, for his support, words of encouragement and the different advices on how to approach situations.
* The Director of REALIZE, **Mr. MONETTE** and his staffs for trusting us by given us the opportunity to serve as interns in their prestigious enterprise.
* Our academic supervisor **Mr. MESSIO** for her moral support, effort and advices throughout the internship period.
* To our academic teachers**, Mrs. TCHOUTOUO ISABELLE, Mr. NGUH PRINCE, Mrs. ONGUENE VANESSA AND Mr. AGBOR ANDERSON** for their advices 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.
* For my sisters for their love and support.
* The countless contributors of open-source programming community, for their great help in learning basic skills and detecting and solving bugs.
* Tomy mother and father, and uncles **NDONLAP, MBANYA, AND KAMWA** for her moral and material support.
* A special thank you to **Mr. Wilfried Muiche** for his everyday support towards me.
* To all my classmates for their collaborative work throughout the academic year.

# GLOSARY

* AICS: African Institute of Computer Sciences
* APK: Android Package Kit
* ERD: Entity Relational Diagram
* 2TUP: Two Track Unified Process
* IPA: iOS App Store Package
* IOT: Internet of Things
* MVC: Model View Controller
* UML: Unified Modelling Language
* AI: Artificial intelligence

# ABSTRACT

Technology has revolutionized various aspects of life, playing a crucial role in communication, marketing, and beyond. Web applications and technology, including smartphones, tablets, and laptops, has become an essential part of people's lives. This project, named **"CONCEPTION AND REALIZATION OF A TONTINE MANAGEMENT PLATFORM"** The project aims to revolutionize the traditional tontine financial arrangement by integrating modern technology for enhanced accessibility and efficiency. This application leverages mobile or web technology and a robust backend system to offer users a seamless, real-time interface for managing their tontine contributions and distributions. Through a user-friendly app, participants can easily track their investments, view their standing, and interact with the tontine pool from anywhere. By employing advanced features like real-time notifications, automated calculations, and secure transactions. This project adopts a combination of UML (Unified Modelling Language) and 2TUP methodology to produce a robust and efficient solution for managing and optimizing tontines remotely. By combining UML and the 2TUP methodology, the project aims to deliver a well-designed, user-friendly, and efficient solution for tontine management and optimization. The iterative and collaborative nature of 2TUP, along with the visual modelling capabilities of UML, ensures a structured development process, timely delivery of increments, and a focus on user needs and satisfaction.

**Key-words**

* Tontine Management
* Financial Arrangement
* Contributions
* Distributions
* Automated Calculations
* Secure Transactions
* User Engagement

# 

# RESUME

La technologie a révolutionné divers aspects de la vie, jouant un rôle crucial dans la communication, le marketing, et au-delà. Les applications web et les technologies, y compris les smartphones, les tablettes et les ordinateurs portables, sont devenues une partie essentielle de la vie quotidienne. Ce projet, intitulé "**CONCEPTION ET RÉALISATION D'UNE PLATEFORME DE GESTION DE TONTINE**", vise à révolutionner l'arrangement financier traditionnel de la tontine en intégrant des technologies modernes pour une meilleure accessibilité et efficacité. Cette application utilise la technologie mobile ou web et un système backend robuste pour offrir aux utilisateurs une interface fluide et en temps réel pour gérer leurs contributions et distributions de tontine. Grâce à une application conviviale, les participants peuvent facilement suivre leurs investissements, consulter leur statut et interagir avec le fonds de tontine depuis n'importe où. En utilisant des fonctionnalités avancées telles que des notifications en temps réel, des calculs automatisés et des transactions sécurisées, ce projet adopte une combinaison de UML (Unified Modelling Language) et de la méthodologie 2TUP pour produire une solution robuste et efficace pour la gestion et l'optimisation des tontines à distance. En combinant UML et la méthodologie 2TUP, le projet vise à fournir une solution bien conçue, conviviale et efficace pour la gestion et l'optimisation des tontines. La nature itérative et collaborative de 2TUP, associée aux capacités de modélisation visuelle de UML, garantit un processus de développement structuré, une livraison ponctuelle des incréments et un accent sur les besoins et la satisfaction des utilisateurs.

**Mots clés:**

* Gestion de Tontine
* Arrangement Financier
* Calculs Automatisés
* Contributions
* Distributions

# ABBREVIATION

* 2TUP: Two Track Unified Process
* Admin: Administration
* AICS: African Institute of Computer Sciences

Alphabetic other

* UI: User Interface
* ERD: Entity Relational Diagram
* UC: Usecase
* DBMS: Database Management System
* MVC: Model View Controller
* UML: Unified Modelling Language
* ICT: Information and Communication Technology

**GENERAL INTRODUCTION**

The idea of machines that take-in random facts, process them and send out meaningful information stems from the 11th century, however, computers only saw the light of day in the 20th century. Since then, computers have gone from being the size of an entire room to being able to fit in the human palm and now we even find some as small as a sheet of paper. The tontine is a financial arrangement that originated in 17th-century France and is named after Lorenzo Tonti, the Italian banker who proposed it. During the 18th century, the tontine gained popularity in France and other European countries as a method to finance public projects and government debts, offering a unique way to raise capital while providing income to participants. The African Institute of Computer Sciences (AICS) in fulfilment of its mission to train African engineers in Africa and for Africa puts in place a three-month internship period for level two students. This period gives students the opportunity to put into practice skills acquired in school in a professional milieu and to further develop their problem-solving skills through assistance in companies. We did our internship at “Etech for self-worth”, which is a growing company in the field of computer engineering whose mission is to Bridge the gap between Africa and Technology and form mostly in IOT section. At Etech, we worked on the theme “**"CONCEPTION AND REALIZATION OF A TONTINE MANAGEMENT PLATFORM**”, with the aim of providing a user interface with dynamic management of their funds. The achievement of this theme was carried out under 8 main sections which are;

1. **Insertion Document:** In this book, we shall present the company in which we spent our internship period and the way we were welcomed in the company. At the end of this section, we will elaborate on our theme.
2. **Existing System:** Here, we shall present the already present system in place, that is the one used for consultation and follow-up purposes.
3. **Specification Book:** In this book, we specify the needs of the user taking into considerations the time and cost of the project.
4. **Analysis Document:** Here, we shall present the analysis method chosen together with the presentation of all the diagrams used for the analysis of this project.
5. **Conception phase:** This presents the generic and detailed conception of the project to bring out real world constituents.
6. **The Realization phase:** This phase will permit us to visualize the implementation process of the solution.
7. **Test of functionalities:** In this phase, we shall present to you the different functionalities or modules of our application and how they work.
8. **The User Guide:** This elaborates on all conditions necessary to use the application and how to use it**.**

**PART ONE: INSERTION PHASE**

Preamble

This phase presents the details of how we were integrated in the host company, the company presentation and organization, and a brief introduction to our project.

Overview

INTRODUCTION

1. WELCOME AND INTEGRATION
2. GENERAL PRESENTATION OF REALIZE CENTER
3. ORGANISATION OF THE COMPANY
4. HARDWARE AND SOFTWARE RESOURCES OF THE COMPANY
5. BRIEF PRESENTATION OF THE PROJECT THEME

CONCLUSION

## INTRODUCTION

The insertion phase is a period (generally of 02 weeks) reserve for the different interns to discover and to familiarize with the working environment. Here, we got to know about the staffs, the different hardware and software resources used, the different departments which constitute the enterprise, how the company function both internally and externally and we were introduced to our work space. During this period, we were also attributed an internship master often called professional supervisor and a theme. We also had a time to discuss amongst us interns on topics like what we love doing most, what we dislike, our believes and experiences. We shared about different realizations and failures in life.

## WELCOME AND INTEGRATRION

### **Welcome**

We arrived at REALIZE on Monday July 2023 at 8:00am. We were welcomed, by **Mr. TANUE MONETTE BUKWATI** the enterprise frontend developer, who introduced us to our work space and his collaborators. Later run, he gave us the task to do some findings on the possible internship themes that we can work on.

### **Insertion**

A working day at REALIZE start from 07:30am to 04:30pm. Our tasks in the company generally follow a predefine routine. Every day we were to write five pages of our internship report, to complete the assigned task for the day. It was also an opportunity to present our difficulties and challenges to everyone and receive directives.

## GENERAL PRESENTATION OF THE COMPANY

### **History**

Realize is a Cameroon based non-governmental tech start-up institute founded by Mr. **NDELOGAKEH Daniel** in 2022 which proposes IT solutions and empowers the use on new technologies in Cameroon

### **What is REALIZE all about**

### **Missions**

The mission of Realize mainly relates to the empowerment and the perpetuation of Technology use. These missions include:

* + Design and realize mobile-oriented software for companies and individuals;
  + Assist newborn companies for quick growth using computer sciences;
  + Provide training and certifications to improve qualified human resources in many fields of study;
  + Take an active part in the sustainable development of the world through innovative solutions and virtual reality.

### **Vision**

At Realize, we believe that the true potential of every organization and individual can be unlocked through the harmonious fusion of analysis, development, realization, and data analytics. Our vision propels us forward, guiding us to make a lasting impact on the digital landscape and shape a future where innovation knows no bounds."

### **Activities**

The activities of Realize 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, IT support and innovation

### **Geographical location**



## ORGANISATION OF THE ENTERPRISE

### **Administrative Organization of REALIZE**

REALIZE is administratively organized as follows;

#### **The General Management**

**REALIZE** is a company made up of a team of engineers in the field of computer science. The main mission of the company is to solve the computer problems that companies face on a daily basis in the following areas:

* Mobile development
* Web development
* Artificial Intelligence
* Data analysis
* Big data

• Machine Learning

### **Human resource department**

This department is in charge of the following:

* + Recruitment and Hiring
  + Acts as a liaison between employees and management;
  + Manages employee compensation and benefits programs;
  + Develops and implements company policies and procedures, ensuring compliance with employment laws and regulations.
  + Manages administrative tasks related to employee records, data management, and

HR systems

### **Communication Department**

This department is in charge of the following:

* + - Handles public relations activities, which involve managing the company’s reputation and image;
    - Responsible for crafting and delivering messages to external stakeholders such as customers etc.;
    - It fosters effective communication within the company;
    - They create and manages content across different platforms and channels;

### **Department of Financial Affairs**

This department is in charge of the following:

* + - Responsible for developing and managing the company’s financial plans and budgets;
    - Prepare and presents accurate and timely financial reports to management, stakeholders, and regulatory authorities;
    - Responsible for managing cost and expenses within the organization; ➢ Ensures compliance with tax laws and regulations. Etc.

### T**echnical department**

This department is in charge of the following:

* + - Responsible for managing the company’s technological infrastructure, including networks, hardware;
    - Play a role in managing and maintaining the company’s data;
    - Responsible for implementing and maintaining cybersecurity measures to protect the company’s digital assets from potential threats;
    - Interacts with technology vendors and manages relationships with external service providers. Etc.

### **Software Engineering Department**

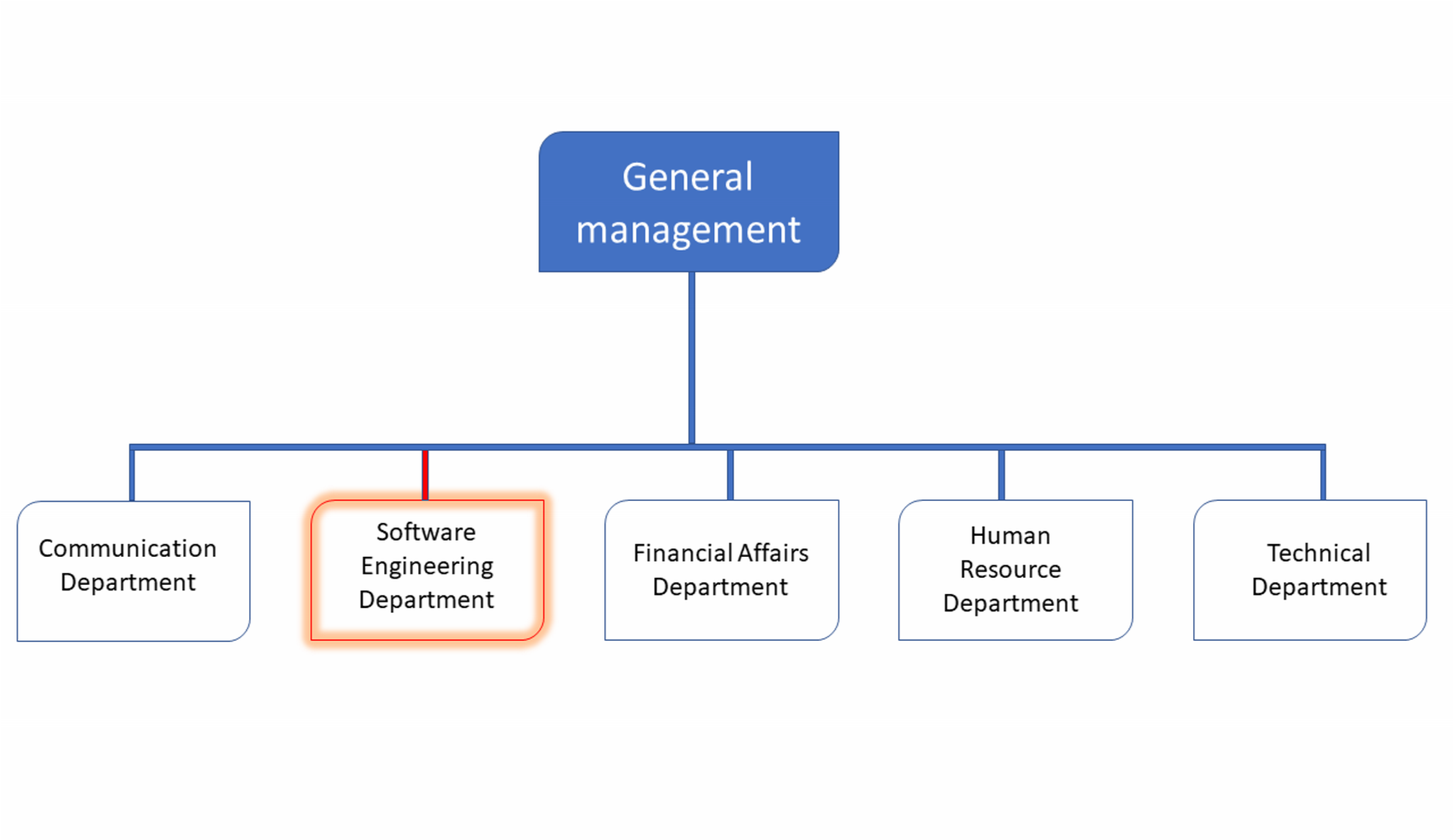
This department is in charge of the following:

* + - Primarily responsible for developing software applications and systems;
    - Responsible for maintaining and supporting software application throughout their lifecycle;
    - Evaluation and realization of projects etc.

### **Functional organization of REALIZE**

The functional branch of REALIZE is organized as follows; The organizational chart is first and foremost a communication tool designed to facilitate understanding of the relationships and connections within the company. It allows for a global view of the company in terms of services, divisions, and much more.

**REALIZE** has four divisions within it, and we have been assigned to all divisions except the secretariat.



*Figure*

*2*

*:*

*Company functional organization (source: Realize*

*)*

### **Attributions**

### **Web Programming Division:**

In this division, all web development projects of the company, partners and clients are routed for realization. Its role is to:

* Design and produce the technical specifications;
* Respect and enforce the application of company standards;
* Write the content of web programming training and provide training in the context of training.
  + 1. **Mobile Programming Division:**

In this division, all mobile development projects of the company, partners and clients are routed for realization. Its role is to:

* Design and produce the technical specifications;
* Respect and enforce the application of company standards;
* Write the content of mobile programming training and provide training in the context of training.

### **Secretary:**

For billing, photocopies, and customer registration.

## HARDWARE AND SOFTWARE RESOURCES OF THE COMPANY

### **Hardware Resources**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Designation | Quantity | Characteristics | Observation |
| 1 | MacBook Pro | 1 | APPLE | Good |
| 2 | Modem | 1 | CAMTEL | Good |
| 3 | LCD Screen | 1 | HP | Excellent |
| 4 | Training equipment and office furniture | / | / | / |

**Table 1**: Hardware resources of REALIZE (source: MERCURIAL 2023)

### **Software Resources**

As an enterprise specialized in IT, REALIZE also comprises of software resources such as;

**Table 2**: Software resources of REALIZE (source: MERCURIAL 2022)

|  |  |
| --- | --- |
| Designation | Software |
| Operation system | Windows 10, Mac OS, parrot Linux |
| Design tools | Photoshop |
| Integrated development environment (IDE) | Visual studio code, |
| Text editor | Sublime text, notepad++ |
| Database management system (DBMS) | MongoDB, PostgreSQL, MySQL |
| Web browser | Google chrome, Microsoft edge |
| Document editor | Microsoft office word |
| Presentation | Microsoft office PowerPoint |

## BRIEF PRESENTATION OF THE PROJECT THEME

During our insertion phase in REALIZE, we were asked to look for themes which solves a specific problem and something innovative. Professional supervisor took some time to look into what we proposed as themes and he also proposed some themes ideas. Finally, we had to go for **“CONCEPTION AND REALIZATION OF A TONTINE MANAGEMENT PLATFORM”**. This system is a software that is made up of a web application for users. With this application, users will have the possibility to contribute their money easily and more efficiently where records are been kept and also reduce manual paper work.

**Conclusion**

The insertion phase was a very educative one, we got to know more about the history of the company, how it all started till the level it is right now, we got to know the staff and other interns, in fact we felt at home. Finally, we decided to work on the theme **“CONCEPTION AND REALIZATION OF A TONTINE MANAGEMENT PLATFORM”.** Now that we have chosen the theme we will work on; we can move on to the specification book of our report where we will identify the objectives and requirements for our project.

**PART TWO: EXIXTING SYSTEM**

### Preamble

The existing system is a part of the internship report where we shall give a detailed explanation of our theme. That is, what the theme is all about and also to give more precision concerning the application we are to produce. Also, it provides a deep understanding of the system currently in placed associated to the various limitations, the problems that result from these and the solution we propose.

### Overview

INTRODUCTION

1. PRESENTATION OF THE THEME
2. DESCRIPTION OF THE EXISTING SYSTEM
3. CRITICISMS OF THE EXISTING SYSTEM
4. PROBLEMATIC
5. PROPOSED SOLUTIONS

CONCLUSION

# INTRODUCTION

In order to successfully achieve a project, we need to have clear information of what the project will consist of, this is where the existing system part of our report comes into play.

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 the current one. After studying the existing system, we shall present our different critics concerning it. We shall also give the problematic as well as the propose solution to these problems.

## PRESENTATION OF THE THEME

During our first days at Realize, we were encouraged by our professional supervisor to take a day or two to come up with a suggestion of a theme for our internship period. Taking into consideration the desire of the company to work on innovative projects as expressed by the companies objectives

Chief Executive Officer during his welcome address to us, and our desire to contribute in our humble way to solving day-to-day problems, we proposed the theme “**CONCEPTION AND REALIZATION OF A TONTINE MANAGEMENT PLATFORM**”. A tontine management app system plays a crucial role in automating and streamlining the administration of tontine investments—a financial arrangement where participants pool funds and receive payouts based on their longevity or other agreed-upon terms. The app helps manage contributions, track participant records, and administer payouts efficiently, ensuring transparency and accuracy. It often includes features like automated calculations of payouts, notifications, and detailed reports, reducing manual errors and administrative burden, and enhancing the overall experience for participants and administrators alike.

## STUDY OF THE EXISTING

Many tontines today face a range of significant challenges that impact their effectiveness and appeal. One major issue is the regulatory landscape; tontines often operate in a gray area with outdated or inconsistent regulations that can lead to legal ambiguities and difficulties in compliance. In some jurisdictions, tontines may not fit neatly into existing financial or insurance frameworks, creating uncertainty for both administrators and participants. Transparency is another critical concern. Traditional tontines, which rely on manual processes and informal record-keeping, can suffer from a lack of clear communication about terms and conditions, leading to misunderstandings and disputes among participants. The administrative burden associated with tontines is also considerable. Managing contributions, tracking individual participant data, calculating payouts based on longevity or other criteria, and ensuring accurate and timely disbursements require complex and meticulous record-keeping. Without modern, automated systems, these tasks are prone to errors and inefficiencies. Additionally, tontines can be vulnerable to issues of trust and security; participants must rely on the integrity of the managing entity, and any perceived or actual mismanagement can lead to dissatisfaction and disputes. The combination of these challenges makes tontines less attractive compared to other financial products that benefit from clearer regulations, more robust administrative systems, and greater transparency.

## LIMITATIONS OF THE EXISTING SYSTEM

At the end of our collection of information concerning the daily collection process in a microfinance, we find that the system has weakness that should not be overlooked despite their efforts.

# *Table 3: Criticism, consequences and proposed solution*

|  |  |  |
| --- | --- | --- |
| **Problems** | **Consequences** | **Proposed Solutions** |
| 1. Lack of Transparency | - Disputes among members - Erosion of trust - Potential for fraud | - Implement clear and detailed documentation - Use digital platforms for real-time tracking - Regularly audit and review transactions |
| 2. Poor Management and Supervision | - Mismanagement of funds - Potential for loss of capital - Discontent among members | - Appoint experienced managers or supervisory boards - Provide training and establish clear guidelines - Regular performance reviews |
| 3. Inadequate Record Keeping | - Difficulty in tracking contributions and distributions - Increased risk of errors - Disputes over records | - Use digital tools for record-keeping - Maintain accurate and up-to-date records - Implement backup and recovery systems |
| 4. Regulatory Compliance Issues | - Legal penalties and fines - Risk of shutdown - Damage to reputation | - Stay informed about relevant regulations and laws - Consult with legal experts - Ensure compliance with local and national financial regulations |
| 5. Security Concern | - Risk of fraud or theft - Unauthorized access to sensitive information - Loss of funds | - Implement strong security measures (e.g., encryption, multi-factor authentication) - Regular security audits - Educate members about security best practices |

## PROBLEMATIC

**Given that security is paramount in financial transactions, ensuring a simple and explicit process is crucial, especially when the financial well-being of individuals is at stake. Our tontine system must address various challenges such as transparency, management, and record-keeping. With these critical observations in mind, we must ask: How can we streamline the transaction process between contributors and collectors while maintaining robust security and efficiency? Our application aims to provide a solution by integrating advanced security measures and user-friendly features to facilitate secure and efficient financial interactions within the tontine.**

### **PROPOSED SOLUTION**

After our study and criticism of the existing situation, we propose to design a Web application to solve the problems stated aboveby allowing: The application to;

* **Implement Real-Time Reporting:** Provide a digital platform where participants can view real-time updates on contributions, withdrawals, and other transactions.
* **Detailed Records:** Maintain detailed and accessible records of all transactions and decisions made within the tontine.
* **Qualified Management Team:** Appoint experienced and trustworthy individuals to oversee the tontine operations and ensure proper management.
* **Digital Record System:** Utilize a digital record-keeping system to accurately track all transactions, contributions, and withdrawals.
* **Backup and Security:** Regularly back up records and ensure they are securely stored to prevent data loss. Deposit contribution
* **Implement Strong Security Measures:** Use encryption and secure authentication methods to protect digital transactions and sensitive data.

**CONCLUSION**

Having reached the end of this part, we studied the existing system by questioning Mrs. Nono through the survey that enabled us to come out with the limitations of the system that leads to the consequences and we proposed solutions to the various limitations. Finally we proposed our solution which is a web application.This step is necessary because we need to understand how the system put in place currently works before we can facilitate or ameliorate the processes been carried out in this system.

**PART THREE: SPECIFICATION BOOK**

Preamble

The primary goal of the specification book is to outline with great precision the requirements or need of the users and the description of the resources necessary to realize the project. It is considered as a communication and description tool which permits us to avoid inadequate results.

Overview

INTRODUCTION

1. CONTEXT AND JUSTIFICATION
2. OBJECTIVES OF THE PROJECT
3. EXPRESSION OF NEEDS
4. PLANNING OF THE PROJECT
5. ESTIMATION OF COST
6. CONSTRAINTS
7. DELIVERABLES

CONCLUSION

# INTRODUCTION

In order to successfully achieve a project, we need to have clear instructions so as to avoid failure. Hence, realizing the document phase called the specification book will help avoid the failure of our project. The specification book permits to specify the expectations of the customer as well as the standards that will govern the project for a good, smooth, precise and concrete project. It establishes a certain agreement between the client of the project and the person who is supposed to realize the project. It presents the objectives and needs of the system to be developed. The present specification book relates to the context and justification, the needs of the project, the plan and the deliverables.

## I. CONTEXT AND JUSTIFICATION

### **1. Context**

### Tontines, originating in the 17th century and named after the Italian banker Lorenzo de Tonti, were initially investment plans where participants contributed funds to a common pool, with dividends distributed among survivors, thus increasing their shares as participants dwindled. This concept has historically been used not only as an investment vehicle but also as a form of mutual aid society, especially prevalent in parts of Africa and among diaspora communities, where they serve as crucial financial support systems. Traditional tontines are managed through manual record-keeping and often face issues of transparency, trust, and efficiency. With advancements in technology, TontiTrack as a modern online tontine has significantly improved the existing system by digitizing records, automating distributions, and ensuring transparent operations through decentralized ledgers, technology can enhance security, reduce fraud, and streamline operations, making tontines more accessible and reliable. This technological integration not only revitalizes an age-old system but also aligns it with contemporary needs for financial security and community-based financial solutions.

### **2. Justification**

Tontines have long served as vital community-driven financial tools, offering members the opportunity to pool resources for mutual benefit. In many parts of the world, particularly in Africa and among immigrant communities, tontines provide a lifeline for individuals who may lack access to traditional banking systems. However, the current manual systems used for managing tontines are prone to errors, delays, and fraud, which undermine trust among participants. The lack of transparency and efficiency in managing contributions and payouts often results in disputes and financial losses. By modernizing tontine management through technology, such as secure digital platforms, automation, and a wallet for transparency, we can address these weaknesses. The use of smart contracts ensures that contributions are accurately tracked, payouts are timely, and all transactions are immutable and verifiable. This technological shift not only preserves the communal and supportive nature of tontines but also instills confidence and security, providing participants with an improved, reliable, and scalable financial solution.

## II. OBJECTIVES OF THE PROJECT

### **1. The general objective**

The general objective of the tontine management application is to create a platform that streamlines the process of contribution collection and distribution in a secure and efficient manner, while providing participants with transparent access to their financial records, both past and present, ensuring accountability and trust among members.

### **2. Specific Objectives:**

The specific objectives of this project are to:

* **Automate Contribution Tracking:** Implement a system to automatically record and track each participant's contributions, ensuring accuracy and eliminating manual errors.
* **Ensure Secure Transactions:** Utilize encryption and secure payment gateways to guarantee that all financial transactions within the platform are protected from fraud and unauthorized access.
* **Provide Real-Time Financial Records:** Allow users to view their financial history, including past contributions and payouts, in real-time, with detailed reports accessible at any time.
* **Facilitate Timely Payouts:** Automate the distribution process to ensure that payouts are made promptly and in accordance with tontine rules.
* **Simplify Group Management:** Enable tontine managers to easily create, manage, and monitor tontine groups, including setting contribution schedules and payout cycles.

## III. EXPRESSION OF NEEDS

### **1. Functional Needs:**

The functional needs are the needs for which our application should be able to perform in other to carry out the assigned tasks. Thus, the platform that we will set up will have to include the following functionalities and modules:

* Registration and Login
* Profile Management
* Create and Manage Tontine Groups
* Contribution Scheduling
* Member Invitations
* Automated Collection Process
* Contribution Confirmation
* Missed or Late Contributions
* Automated Payouts
* Payout Notifications
* **View Financial Records:** Provide users with access to both past and present financial records, including contributions, payouts, and balances.
* **Generate Financial Reports:** Generate detailed reports summarizing the financial activity within the tontine group.
* **Secure Payment Processing:** Integrate secure payment gateways for contribution collection and payout processing.
* **Contribution Reminders:** Send reminders to users for upcoming or due contributions.
* **Payout Notifications:** Notify users when payouts are scheduled or completed.
* **Alerts for Delays or Issues:** Alert members and managers if there are any issues with contributions, payouts, or group activities.
* **Responsive Web Interface:** Ensure the web application is responsive and accessible from mobile and desktop browsers.

### **2. Non-Functional Needs**

#### Hardware Requirements

**Quality Attributes**

* **Portability and Ease of Installation**: The system should be accessible via lightweight, portable devices such as smartphones, tablets, and laptops. The software installation process should be straightforward, requiring minimal technical skills. Users should be able to quickly set up and start using the application without requiring expert help.
* **Reusability and Reliability**: The platform should be flexible and reusable across various tontine groups, whether personal, community-based, or enterprise-level. It must reliably record and manage financial transactions and be trusted for accurate financial tracking and reporting.

**Safety Requirements**

* The system should employ strong security protocols to protect user data and ensure safe handling of financial transactions. No unauthorized access or malicious activity should be able to affect the application or its processes, and secure payment channels must be guaranteed to avoid financial losses or fraud

#### Software Requirements

**Security Requirements**

* The software must have robust security measures to protect user data, including end-to-end encryption, secure authentication (e.g., multi-factor authentication), and the safeguarding of financial transactions. Data integrity must be guaranteed, ensuring that no data is altered during storage, retrieval, or transmission. Privacy of user financial records must also be maintained.

**Performance Requirements**

* The application should have a user-friendly interface that allows users to easily manage tontine groups, view records, and make transactions without technical challenges. The software must have quick response times and efficient connection to backend services. It should scale well as user numbers grow, allowing multiple users to interact with the system simultaneously without degradation in performance.

**Scalability and Availability**

* The platform should support an increasing number of users and tontine groups without a significant drop in performance. It should be available across multiple devices and ensure high availability (uptime) so users can access their tontine data anytime and anywhere.

**Maintainability and Flexibility**

* The software should be easy to maintain, with regular updates for bug fixes, security patches, and feature enhancements. It should also be flexible to accommodate future changes, like adding new features or integrating with third-party financial tools, without major disruption.

**Usability and Accessibility**

* The application must be intuitive and easy to navigate for users of varying technical expertise. It should also be accessible to people with disabilities by supporting common accessibility standards like screen readers and responsive design, ensuring a smooth experience across different device screen sizes.

## IV. PROJECT PLANNING

#### **Chronogram of activities**

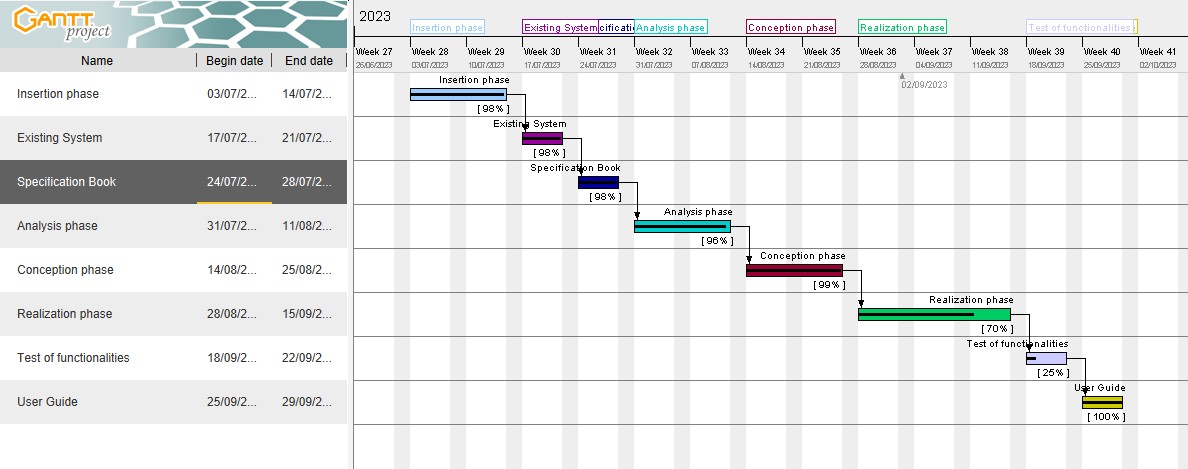
##### *Table 4: Planning of the project*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PHASE** | **OBJECTIVE** | **OUTPUT** | **DURATION** | **PERIOD** |
| **INSERTION** | Collection of information on the enterprise | Insertion book | 2 weeks | 03rd July to 14th July |
| **EXISTING SYSTEM** | Study of the existing system | Existing System | 1 week | 17th July to  21st July |
| **SPECIFICATION BOOK** | Specification of the user needs | Specification Book | 1 week | 24th July to 28th July |
| **ANALYSIS** | Capture of needs Use case and textual description  Modelling | Analysis Book | 2 weeks | 31st July to 11th August |
| **CONCEPTION** | Preliminary conception and  Detailed conception | Conception book | 2 weeks | 14th August to 25th  August |
| **REALIZATION** | Implementation Unitary test  Integration Test  Development,  Deployment,  Component diagrams | Realization book | 3 weeks | 28th August to 15th  September |
| **TEST OF FUNCTIONALITIES** | Testing of the software and debugging | Test of functionalities | 1 week | 18th  September to 22nd  September |
| **INSTALLATION AND USER GUIDE** | Documenting software | User Guide | 1 week | 25th  September to 29th  September |

### **Gantt Project planning**

**b**

**Gantt Project planning**



*Figure*

*3*

*:*

*Gantt planning*

## V. ESTIMATION OF COST

### **1. Software Resources**

Table 5: Software resources of this project (source: Mercurial 2021)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RESOURCES** | **DESIGNATION** | **USAGE** | **QUANTITY** | **UNIT COST**  **(FCFA)** |
| **Text Editor** | Microsoft Office  2019 | Used for the creation of our report and  PowerPoint | 1 | **500,500** |
| **Web** |  |  |  |  |
| **browser** | Google Chrome | View web pages | 1 | **Free** |
| **Code Editor** | Visual Studio  Code | For writing the code of the application | 1 | **Free** |
| **Project planning** | Gantt Project | For building a  Gantt chart | 1 | **Free** |
| **Geo-location** |  |  |  |  |
| **plan**  **designing tool** | Ichogram | To draw the location plan | 1 | **Free** |
| **UML**  **Analysis** | Visual paradigm  (Community edition) | Modeling tool | 1 | **Free Trial** |
| **TOTAL 1** |  |  | **8** | **500,500** |

### **2. Hardware Resources**

Table 6: Hardware resources of the project (source: Mercurial 2023)

|  |  |  |
| --- | --- | --- |
| **Material** | **Quantity** | **UNIT COST (FCFA)** |
| **LAPTOP HP**  **Intel Core i3** |  |  |
| **500 GO, RAM 4**  **GO,** | **1** | **250,000** |
| **Printers** | **1** | **500,820** |
| **Local network** |  |  |
| **installation** | **1** | **300, 000** |
| **8GB USB Key** | **1** | **6325** |
| **Smart phone** | **2** | **150,000** |
| **Total2** | **6** | **1,207,145** |

### **3. Human Resources**

Table 7: Human Resources of the project

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ROLE** | **NUMBER OF DAYS** | **QUANTITY** | **COST PER DAY** | **TOTAL PRICE (FCFA)** |
| **Project**  **Manager** | **90** | **01** | **30 000** | **2 700 000** |
| **Analyst** | **21** | **01** | **25 000** | **525 000** |
| **UI/UX Designer** | **07** | **01** | **20 000** | **140 000** |
| **Programmer** | **30** | **01** | **15 000** | **450 000** |
| **Tester** | **14** | **02** | **10 000** | **140 000** |
| **Margin error** | **/** | **/** | **/** | **1,475,000** |
| **TOTAL3** |  |  |  | **4 840 000** |

### **Global estimation**

Table 8: Global estimation

|  |  |  |  |
| --- | --- | --- | --- |
| **TOTAL 1(FCFA) TOTAL 2(FCFA) TOTAL 3(FCFA) OVERALL TOTAL**  **(FCFA)** | | | |
| **500,500** | **1,207,145** | **4 840 000** | **6,547,645** |
| **SIX MILLION FIVE HUNDRED AND FORTY-SEVEN THOUSAND SIX**  **HUNDRED AND FORTY-FIVE** | | | |

### **List of participants**

Table 9: List of participants

|  |  |  |
| --- | --- | --- |
| **NAME** | **TITLE** | **ROLE** |
| **Mr. TANUE Monette** | Developer at Realize | Professional supervisor |
| **Mr. MESSIO** | Lecturer at AICS-Cameroon | Academic Supervisor |
| **TCHOUGA**  **MICHELLE SERENA** |  |  |
|  | Software engineering student at AICS-Cameroon | Analyst and developer |

1. **CONSTRAINTS**

In every project, it is important to consider and address any limitations and risks to ensure the project's success. Therefore, we have identified the following constraints:

**a. Criterions of Acceptability**

The application will be deemed acceptable if it meets the needs of the users, including the secure handling of financial transactions, transparency in contribution and payout tracking, and ease of use for both contributors and tontine managers. The application must also implement all the specific functionalities outlined in the requirements, including automated contribution management, real-time reporting, and data security.

**b. Time Constraint**

The development of the tontine management application is expected to be completed within 10 weeks, starting from the project’s inception. This timeline includes development, testing, and deployment phases, ensuring the product is ready for launch by the specified deadline.

**c. Budgetary Constraint**

The development budget for the project will be capped at $15,000, which covers the cost of development, testing, and deployment. Any additional costs, including unforeseen expenses, must be carefully managed to remain within this budget, with priority given to essential features and security.

1. **DELIVERABLES**

Regarding the project, below lies its deliverables

A report composed of the following documents

* The insertion document.
* The Existing System
* The specification book.
* The analysis phase.
* The conception phase.
* The realization phase. ❖ Test of functionalities.
* The user guide.

**CONCLUSION**

The specification book helps us to site the different needs we need in order to implement our application for tontine management, we saw list of participants and deliverables for our project. In time allocated for this section, we were able to accomplish the different task, we will move directly to the next phase which is the analysis phase. In the analysis phase we will study the existing system in detail and model our system with a modelling language and process.

**PART FOUR: ANALYSIS PHASE**

Preamble

After specification book, we have the Analysis phase which permits us to represent a detailed analysis of the limitations identified in our context, and our solution, through a software development process and modelling language.

Overview

INTRODUCTION

1. COMPARATIVE STUDIES OF UML AND MERISE
2. COMPARATIVE STUDY OF UNIFIED PROCESSES
3. JUSTIFICATION OF MATHOD OF ANALYSIS
4. MODELING OF THE PROPOSED SOLUTION

CONCLUSION

# **INTRODUCTION**

The analysis book permits us to examine in an explicit way the existing system, its limitations and how we can remedy them. We will also describe in details the modeling language known as UML (Unified Modeling Language) which is coupled with Two Tract Unified Process (2TUP) to form a method and its justification why we decided to use it in preference of another. Then we will dive directly into the modeling of the proposed solution consisting of diagrams that meets the requirements of the functional needs.

## COMPARATIVE STUDY OF UML AND MERISE

1. MODELING WITH MERISE (2.5)

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.

* 1. MODELING WITH UML (2.5)

The unified modeling language (UML) is a general purpose, developmental modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

The Unified Modeling Language (UML) was standardized in January 1997 by the Object Management Group (OMG) which is an American association created in 1989 and aims to promote and standardize the object model in all its forms. In 2005, UML was also published by the international organization for standardization (ISO) as an approved ISO standard. Uml since 2015 is in its version 2.5. This version consists of fourteen diagrams classified into structural and behavioral diagrams

### Structural Diagrams:

Structural diagrams represent the static components of a system; they emphasize on what should be in the system we are modeling. They include:

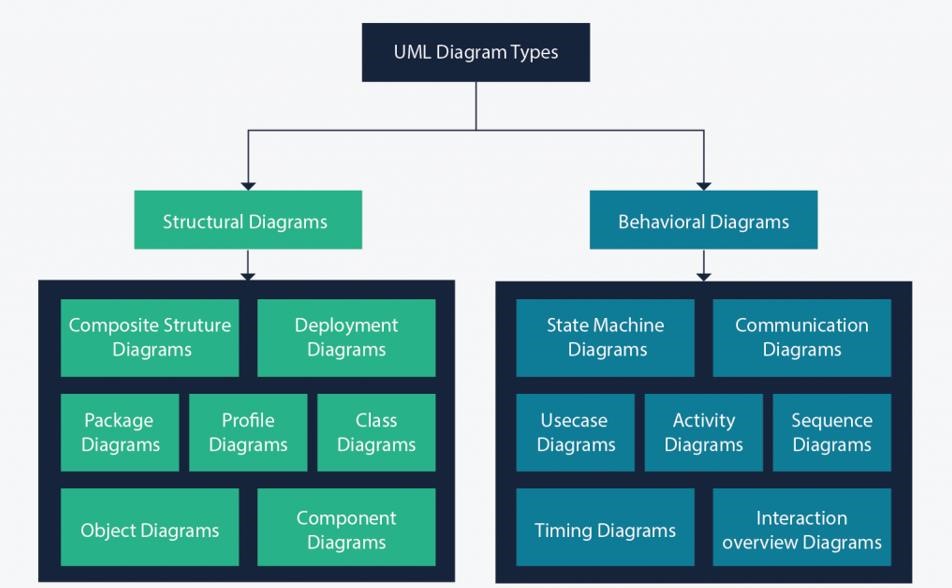
* Class diagram;
* Object diagram;
* Package diagram;
* Composite structural;
* Deployment diagram;
* Component diagram;
* Profile diagram.

### Behavioral Diagrams:

The behavioral diagrams capture the dynamic state of a system; they emphasize on what should happen in the system we are modeling. They are:

* + - Use case diagram;
    - Activity diagram;
    - State machine diagram;
    - Sequence diagram;
    - Communication diagram;
    - Interaction overview diagram;
    - Timing diagram

1. **UML 2.5 diagrams overview:**



##### Figure 4 : UML 2.5 diagrams overview

##### (Source: https://creately.com/blog/diagrams/uml-diagram-types-examples/)

It is important to note that uml is a modeling language and not a method or procedure. Hence, 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. A unified process is a generic method for developing software. This implies it is necessary to adapt the UP to the context of the project, team domain and or the organization. We will use the Two Tracks Unified Process (2TUP) throughout our project to implement our solution.

###### **Table 12: Differences between UML and MERISE**

|  |  |
| --- | --- |
| 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 |

## COMPARATIVE STUDY OF UNIFIED PROCESSES

A process can be defined as a partially sequence of steps that permits us to obtain software systems or evolution of an existing one. The main objective of software development is the production of quality software that response to the needs of the users during a particular time and at a particular cost.

#### 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.

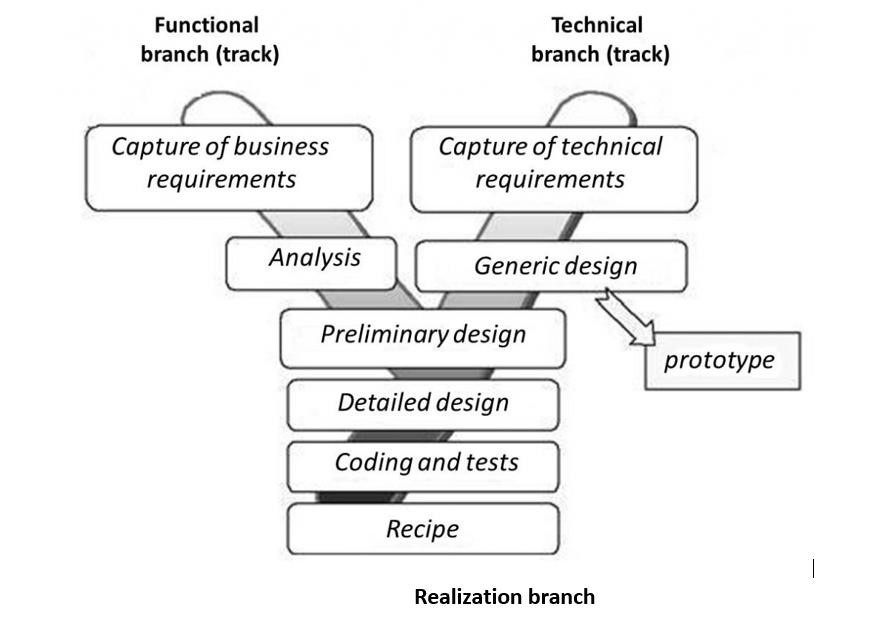
**An increment** is the difference between two released products at the end of two iterations. Each iteration that the group is capable of integrating the technical environment in order to develop a final product and give users the possibility of having tangible results.

**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.

#### The Two Track Unified Process (2TUP)

2TUP is a unified process which is belt 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-sharped 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 bellow illustrates the branches of 2TUP.

******

###### ***Figure 5: 2TUP diagram***

###### (Source:https://www.mysciencework.com/omniscience/pervasive-mobile-healthcare-system-basedon-cloud-computing)

##### **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

##### **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. The different diagrams are shown in the table below.

##### **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.

**Detailed conception:** This is the detailed design of each feature of the system.

**Coding and testing:** This are 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.

## JUSTIFICATION OF METHOD OF ANALYSIS

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

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.

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.2TUP is centered around the creation and maintenance of a model, rather than the production of mountain documents.

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.

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. 2TUP by permitting the project team identify and test the key functionalities of the system limits the risk related to building the system.

## MODELING OF THE PROPOSED SOLUTION

### **Capture of Functional Branch**

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.

#### **Use Case Diagram:**

##### **Definition:**

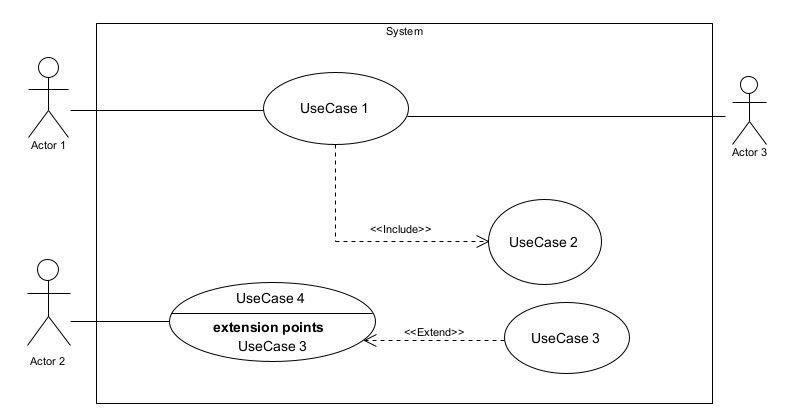
A use case diagram shows the functionalities of a system, their interdependencies and how they relate with actors of the system. A use case is a specification of behavior.

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.

##### **Formalism:**

**Figure 6: Use case diagram formalism**

**Table 13: Use case diagram component**

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

##### **The Actors of Our System**

**Table 14: Actors of our system.**

|  |  |  |
| --- | --- | --- |
| **Actor** | | **Role** |
| **Member** | Can make contributions, receive payouts, receive sanctions and view reports | |
| **Secretary** | Classifies reports | |
| **President** | Create a tontine group | |
| **Admin** | Manage user’s accounts | |

##### **Textual description of Authentication**

**Table:15 Textual description of Authentication**

|  |  |
| --- | --- |
| Title | Authenticate |
| Summary | The user needs to authenticate |
| Actors | Members, Secretary, President Admin |
| Precondition (s) | 1. The actor has an account on the platform. 2. The app is launched. 3. Has data connection for internet |
| Triggers | The user clicks on authentication button |
| Nominal Scenario | 1. The system displays the login form. 2. The actor fills and submits the form. 3. The system verifies conformity of the form fields 4. The system sends the data to the DBMS. 5. The DBMS returns result of the query. 6. The system displays a success message to the actor. |
| Alternative Scenarios | 1. At step iv) of the nominal scenario, the user enters mismatched or missing information 2. The system displays an error message then returns to step 2 of the nominal scenario. |
| Post condition of success | The user has access to his/her dashboard |
| Post condition of failure | The user does not have access to the platform |
| Non-functional requirement | Entering the password must not be visible on the screen |

##### **Textual description of Make contributions**

**Table: 16 Textual description of make contributions**

##### **General Use Case Diagram**

**Figure 7: General Use Case Diagram**

##### **Make contributions specific use case**

**Figure 8: Make contribution**

##### **Create and join tontine specific use case**

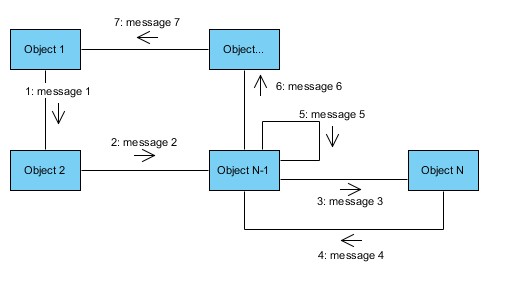
**Figure 9: Create account**

#### **Communication Diagram**

##### **Definition**

It is a diagram which is used to show the relationship between the actors of a system, both the sequence and the communication diagrams represent the same information but differently. Instead of showing the flow of message. It depicts the architecture of the object residing in the system as it is based on object-oriented programming.

##### **Formalism**

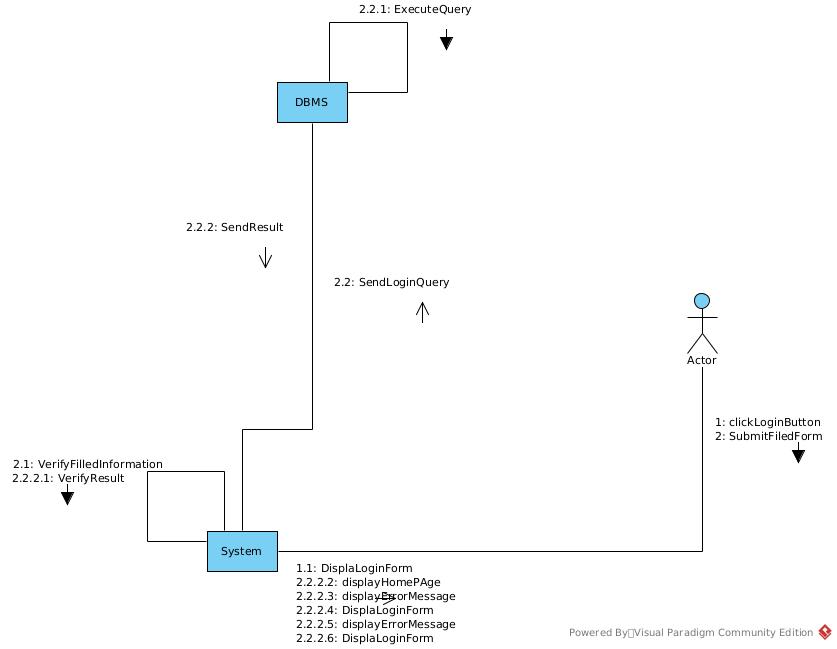


**Figure 10: Communication Diagram Formalism**

**Table: 17 Communication Diagram Component**

|  |  |  |
| --- | --- | --- |
| **Element** | **Representation** | **Description** |
| **Object** |  | An actor represents an individual participant in the interaction conversation. |
| **link** |  | It initiates an association it connects two objects together for them to communicate. |
| **Actor** |  | A role play by an entity that interacts with the subjects. |
| **message** |  | Defines a particular communication between lifelines in an interaction. |

##### **Authentication Communication Diagram**



**Figure 11: Authentication communication Diagram**

##### **Make contribution Communication Diagram**

**Figure 12: Make contribution communication Diagram**

##### **Create and join tontine Communication Diagram**

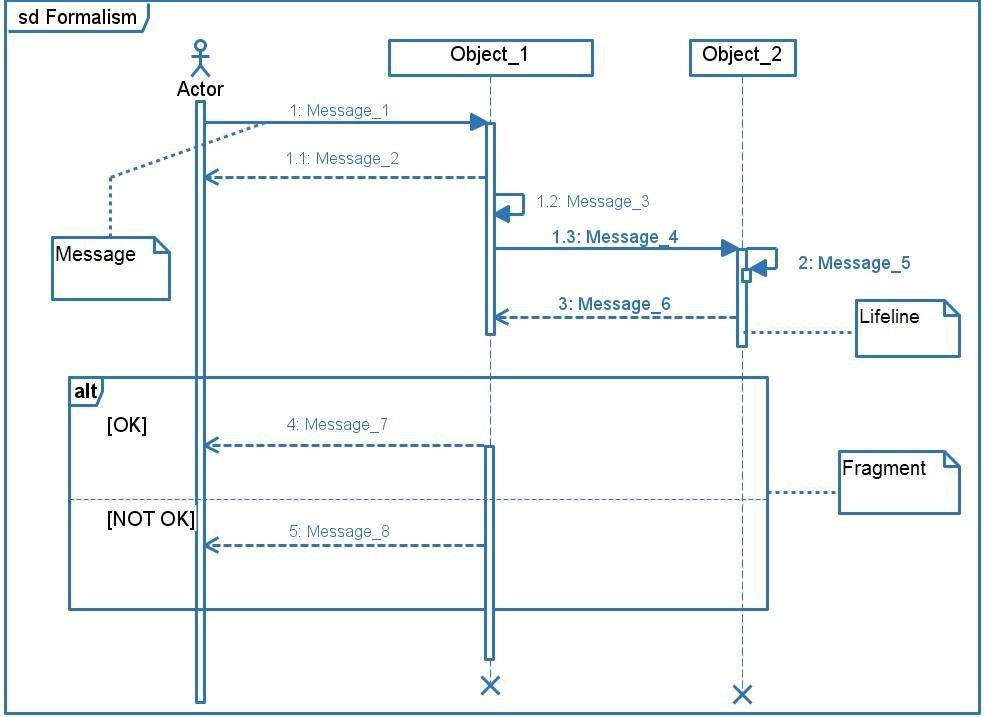
**Figure 13: Choose tontine communication Diagram**

#### **Sequence Diagram:**

##### **Definition:**

A sequence diagram is an interaction diagram which represents the flow of message between elements in a system, it is termed as an event diagram. It portrays the communication between any two lifelines as a time-ordered sequence of events.

##### **Formalism:**



**Figure 14: Sequence diagram formalism**

###### **Sequence diagram components**

**Table 18: Sequence Diagram Components**

|  |  |  |
| --- | --- | --- |
| 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. |
|  |  |  |

###### **Authentication sequence diagram**

**Figure 15: Authentication sequence diagram**

###### **Make contribution sequence diagram**

**Figure 16: Make contribution sequence diagram**

###### **Create and join tontine sequence diagram**

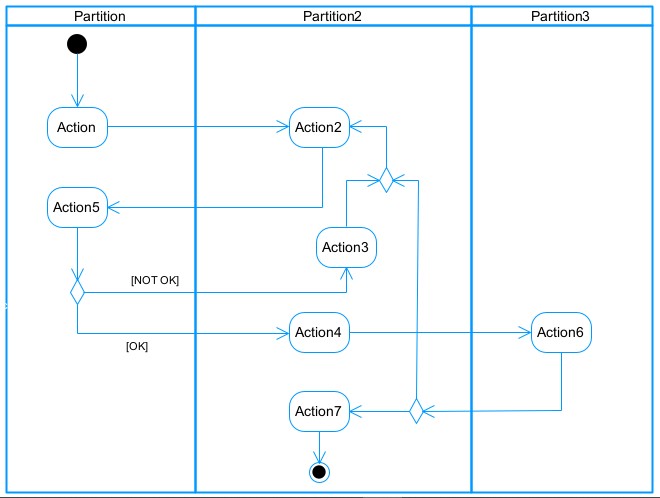
**Figure 17: Make contribution sequence diagram**

#### **Activity Diagram**

##### **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 possible 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.

##### **Formalism**



**Figure 17: Activity diagram formalism**

###### **Activity diagram components**

**Table 18: Activity Diagram Components**

|  |  |  |
| --- | --- | --- |
| **Elements Diagrammatic Representation** | | **Description** |
| **Activity** |  | Use 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 slits 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. |

###### **Authentication activity diagram**

**Figure 18: Authentication activity diagram**

###### **Make contribution activity diagram**

**Figure 19: Make contribution activity diagram**

###### **Create tontine activity diagram**

**Figure 20: Create tontine activity diagram**

###### **Join tontine activity diagram**

**Figure 21: Join tontine activity diagram**

## CONCLUSION

In the analysis phase, we transitioned to focus on the technical and realization branches of the 2TUP methodology. The technical branch outlines the system architecture, a micro services approach to ensure scalability, and a well-structured database design to handle user data and financial transactions securely. We chose a robust technological stack with web-based front-end interfaces and secure cloud-based back-end services, incorporating security measures like encryption and multi-factor authentication to safeguard financial data. In the realization branch, we prioritized modular coding for flexibility, established a comprehensive testing strategy to cover unit, integration, and user acceptance tests, and laid out a deployment plan using cloud infrastructure for scalability and high availability. This phase sets the foundation for successful implementation and ongoing system maintenance, ensuring the application meets both user and technical requirements.

**PART FIVE: 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 shows the link between the analysis phase and the realization phase.

Overview

INTRODUCTION

1. TECHNICAL BRANCH
2. GENERIC DESIGN
3. CAPTURE OF TECHNICAL NEEDS
4. RELATED UML DIAGRAMS .

CONCLUSION

# **INTRODUCTION**

The analysis book permits us to examine in an explicit way the existing system, its limitations and how we can remedy them. We will also describe in details the modeling language known as UML (Unified Modeling Language) which is coupled with Two Tract Unified Process (2TUP) to form a method and its justification why we decided to use it in preference of another. Then we will dive directly into the modeling of the proposed solution consisting of diagrams that meets the requirements of the functional needs.