RÉPUBLIQUE DU CAMEROUN

Paix-Travail-Patrie



ETECH FOR SELF WORTH

Travail-Innovation-Perseverance

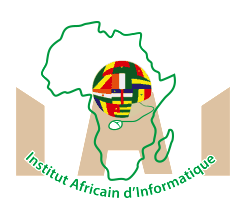
Tél : (237) 673 55 22 37

Site Web : www.etech-sw.org

E-mail : info@etech-sw.org

REPUBLIC OF CAMEROON

Peace-Work-Fatherland



AFRICAN INSTITUTE OF COMPUTER SCIENCE "Centre of Technological Excellence Paul Biya"

BP: 13719 Yaoundé, Cameroon

Tel: 242 672 99 58 / 242 672 99 57

Website: [www.iaicameroun.com](http://www.iaicameroun.com)

Email: contact@iaicameroun.com

Academic Supervisor

**Mr. MESSIO Ulrich**

Lecturer at AICS-Cameroon

Academic Year 2022-2023

Professional Supervisor

**Mr.AGUEGUIA Raoul**

DG of Etech for Self-Worth

Internship carried out from 3rd June to 30 September 2023

In view of obtaining **Higher Technician Diploma (HTD)** in Computer Sciences

Option: **Software Engineering**

Written by

**ESSONO JORDSN RYAN**

Level II Software Engineering Student

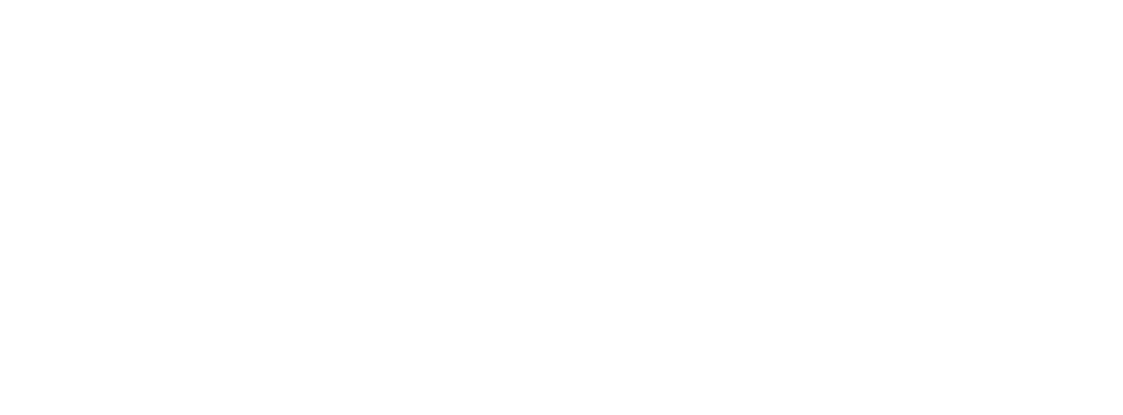
**CONCEPTION AND REALIZATION OF A DIGITAL SMART EGG INCUBATOR**

**THEME:**

INTERNSHIP REPORT

# DEDICATION

# ACKNOWLEDGEMENT



**TO**

**THE FAMILIES**

**ESSONO AND MESSA**

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 AICS-Cameroon, **Mr. ARMAND Claude Abanda**, for his support, words of encouragement and the different advice on how to approach situations.
* The Director of Etech for Self-worth , **Mr. AGUEGUIA Raoul** and his staffs for trusting us by given us the opportunity to serve as interns in their prestigious enterprise.
* Our academic supervisor **Mr. MESSIO ULRICH** for his moral support, effort and advice throughout the academic year.
* To our academic teachers **Mr. AGBOR ANDERSON, Mrs. EBENYE AND Mrs. TCHINGA ALICE** 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 our Professional supervisor **Mr.** **AGUEGUIA RAOUL** and all the members of **ETECH-CENTER.**
* A special thanks to the AICS student’s committee President, **Mr. TSAYOUTSOP KANOU EDSON** for his moral and material support.
* A special thanks tomy brothers and sisterswho helped me all through this project from beginning to the end and who even gave me the idea of the project
* A special thanks to my close friend **Ms**. **MENGUE Rosalie Lauranne.**
* To **The Entire Essono and Messa family** for their moral and material support.
* To **Mrs. ESSONO JACQUELINE AND Mr.ESSONO JEAN-JACQUES** for their moral and material support.
* To all my classmates for their collaborative work throughout the academic year.

# SUMMARY

[DEDICATION 2](#_Toc147908635)

[ACKNOWLEDGEMENT 2](#_Toc27892848)

[SUMMARY 4](#_Toc1385158604)

[LIST OF FIGURES 5](#_Toc502374066)

[LIST OF TABLES 6](#_Toc627873889)

[GLOSSARY 7](#_Toc2111182240)

[ABSTRACT 8](#_Toc1109234057)

[RESUME 9](#_Toc262508664)

[GENERAL INTRODUCTION 10](#_Toc2032989340)

[PART ONE: INSERTION PHASE 11](#_Toc1343275028)

[PART TWO: TECHNICAL PHASE 23](#_Toc1081788182)

[I. EXISTING SITUATION 24](#_Toc206731583)

[II. CONTEXT AND JUSTIFICATION 30](#_Toc2121480716)

[IX. ANALYSIS PHASE 45](#_Toc1718148558)

[III. CONCEPTION PHASE 66](#_Toc40973042)

[IV. REALISZATION PHASE 76](#_Toc1050812889)

[III. FUNCTIONAL TESTING 88](#_Toc946566153)

[IV. INSTALLATION AND USER GUIDE 89](#_Toc1565865447)

# LIST OF FIGURES

# LIST OF TABLES

# GLOSSARY

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

# ABSTRACT

This end-of-year project focuses on the development of an IoT-based egg incubator system integrated with a mobile application. The aim is to create an automated and user-friendly solution for “**DIGITAL SMART EGG INCUBATOR”** using an ESP32 micro-controller, an LCD for display, three fans controlled by a relay module, and a bulb controlled by another relay module.

The system incorporates sensor modules such as the AM2320 for temperature and humidity detection, MQ-135 for ammonia gas detection, and a PIR sensor for motion detection. The mobile application complements the hardware setup by providing a platform for users to create accounts, add and manage multiple incubators, select incubation parameters for different animal types, and monitor the incubation process in real-time.

Additionally, the app enables users to track incubator data, including temperature, humidity, gas levels, water levels, and inclination, through visual representations such as graphs. Users can stop the incubation process at any time, and after completion, they can input the initial number of eggs, good eggs, and bad eggs for statistical analysis.

The project employs tools such as UML, React Native, Python, Django, Google API, and SQLITE3 for efficient implementation. Through comprehensive testing and iterative development, the project aims to deliver a robust and user-friendly IoT-based egg incubator system with a mobile app interface, providing convenience and accurate monitoring for users in the field of egg incubation.

# RESUME

Ce projet de fin d'année se concentre sur le développement d'un système d'incubateur d'œufs basé sur l'IoT intégré à une application mobile. L'objectif est de créer une solution automatisée et conviviale pour un "**INCUBATEUR D'ŒUFS INTELLIGENT NUMÉRIQUE"** en utilisant un microcontrôleur ESP32, un écran LCD pour l'affichage, trois ventilateurs contrôlés par un module relais et une ampoule contrôlée par un autre module relais.

Le système intègre des modules de capteurs tels que l'AM2320 pour la détection de la température et de l'humidité, le MQ-135 pour la détection de gaz d'ammoniac et un capteur PIR pour la détection des mouvements. L'application mobile complète la configuration matérielle en fournissant une plateforme permettant aux utilisateurs de créer des comptes, d'ajouter et de gérer plusieurs incubateurs, de sélectionner les paramètres d'incubation pour différents types d'animaux et de surveiller en temps réel le processus d'incubation.

De plus, l'application permet aux utilisateurs de suivre les données de l'incubateur, notamment la température, l'humidité, les niveaux de gaz, les niveaux d'eau et l'inclinaison, grâce à des représentations visuelles telles que des graphiques. Les utilisateurs peuvent arrêter le processus d'incubation à tout moment et, une fois terminé, ils peuvent saisir le nombre initial d'œufs, le nombre d'œufs viables et le nombre d'œufs non viables pour une analyse statistique.

Le projet utilise des outils tels que UML, React Native, Python, Django, Google API et SQLITE3 pour une mise en œuvre efficace. Grâce à des tests approfondis et un développement itératif, le projet vise à fournir un système d'incubateur d'œufs basé sur l'IoT robuste et convivial, avec une interface d'application mobile, offrant praticité et surveillance précise aux utilisateurs dans le domaine de l'incubation des œufs.

# GENERAL INTRODUCTION

The Digital Smart Egg Incubator is a revolutionary system that combines IoT technology and a mobile application to automate and streamline the egg incubation process. It replaces traditional manual methods with advanced sensors, an ESP32 micro-controller, and relay modules for environmental control. The mobile application serves as a central hub, allowing users to manage multiple incubators, customize parameters, and monitor the process in real-time. With features like visual data representation, pause and stop controls, and statistical analysis, the system offers convenience, accuracy, and improved outcomes.

The hardware components of the Digital Smart Egg Incubator include an ESP32 micro-controller, an LCD display, and relay modules for fan and bulb control. Sensor modules such as the AM2320 (temperature and humidity), MQ-135 (ammonia gas), and PIR sensor (motion) provide precise environmental monitoring. The mobile application integrates with the system, enabling users to create accounts, add incubators, and monitor parameters in real-time.

The mobile application offers a comprehensive set of functionalities, including visual data representation through graphs, pause and stop controls, and statistical analysis. It allows users to track temperature, humidity, gas levels, water levels, and inclination. The app also enables users to input data on the initial number of eggs, good eggs, and bad eggs for statistical analysis and future improvements.

The development of the Digital Smart Egg Incubator project involves tools like UML for system modeling, React Native for mobile app development, Python and Django for back-end programming, Google API for integration, and SQLITE3 for data management. Overall, the Digital Smart Egg Incubator revolutionizes the egg incubation process by integrating IoT capabilities, intelligent monitoring, and a user-friendly mobile application, providing convenience, accuracy, and improved outcomes for users in the field.

* The insertion phase, here we present the company in which we did our internship, and the integration of the interns into the company;
* The specification book, which identifies the need of the future system users and points out different constraints of the project;
* Analysis phase, here we choose our analysis method and the presentation of all the diagram use for the analysis of the project;
* The conception Phase, which presents the generic and detailed conception of the project to bring out real world constituents;
* The realization phase, which presents the choice of technologies and the technics necessary for the implementation of our solution;
* The user guide, which will present a user friendly and graphical description of each functionality of the application.

# 

# PART ONE: INSERTION PHASE

Preamble

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

Content overview

INTRODUCTION

1. WELCOMING AND INTERGRATION
2. GENERAL PRESENTATION OF ETECH CENTER
3. PRESENTATION OF THE ADMINISTRATIVE AND

ORGANISATIONAL FUNCTIONS OF ETECH CENTER

1. RESOURCES (HARDWARE AND SOFTWARE)
2. ORGANIZATIONAL CHART OF ETECH CENTER

CONCLUSION

## INTRODUCTION

The insertion phase is the period during which a student must become familiar with the

environment in which he will perform his internship. It is also the integration of an individual into

a foreign group or environment where the language, the actions, the way of thinking must be purely

professional.

With the aim of putting into practice what have been thought at the African Institute of

Computer Sciences (AICS) Cameroon office, the school integrate an internship with a duration of

4 months, which is the reason why we integrate HARVEY CONSULTING. The goal been to

convert our theoretical knowledge into practice that can help to provide a solution to a specific

problem.

## WELCOME AND INTEGRATION

1. Welcome:

On Monday, June 3rd, 2023, within the Etech For Self-Worth organization, a meeting was held with Mr. Guiffo Joël, the IoT training manager. Etech For Self-Worth is known for empowering individuals through technology, and Mr. Joël's expertise in IoT played a significant role in the meeting. As the IoT training manager, he is responsible for designing and delivering training programs that equip individuals with the necessary skills for the evolving IoT landscape. The meeting provided participants with a valuable opportunity to learn from Mr. Joël's knowledge in IoT, likely covering topics such as applications, trends, and the impact of IoT on industries and society. Overall, the meeting showcased Etech For Self-Worth's commitment to empowering individuals in the digital era and highlighted the significance of IoT in driving technological advancements.

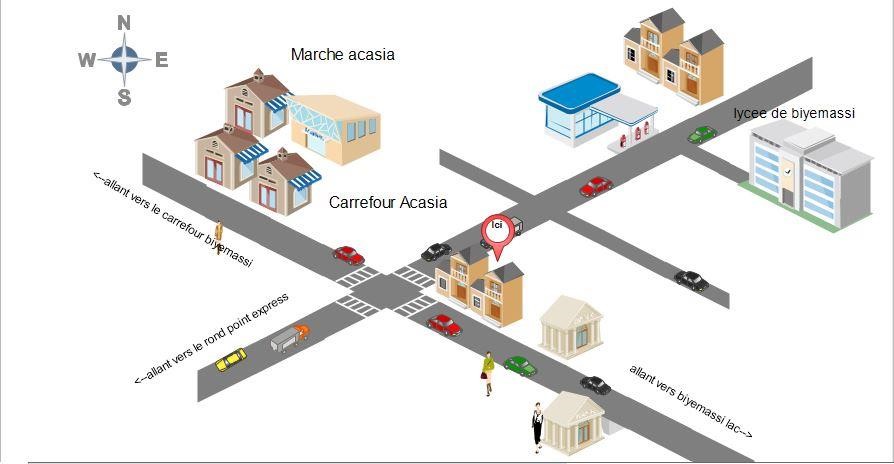
1. Integration:

During a meeting organized by Mr. Guiffo Joel, various important aspects of the establishment were discussed. This included a reminder of internal regulations, the organization's operations, vision, objectives, services, requirements, and the development of the weekly schedule. The meeting emphasized the importance of adhering to conduct guidelines and respecting the current regulations in place. Additionally, valuable advice and recommendations were shared during online sessions with the General Director, Mr. AGUEGUIA LALLAH Raoul, as our professional supervisor was not physically present. Following these discussions, the participants engaged in a fruitful conversation on the designated theme.

## GENERAL PRESENTATION OF ETECH

1. History and Missions:
2. Etech For Self-Worth is the result of a project initiated by Mrs. ZANGUE epse LALLAH Emilienne and Mr. AGUEGUIA LALLAH Raoul on February 12, 2019. The organization was established in compliance with the laws governing Associations in the Republic of Cameroon, specifically law n° 99/014 of December 22, 1999, and law n° 90/053 of December 19, 1990, along with other applicable legal regulations.
3. The founders' vision was to create a company where computer engineers could dedicate their time to undertaking large-scale projects. Additionally, the organization aimed to offer professional training, computer services, and other related activities. This vision materialized on February 12, 2019, with the establishment of Etech For Self-Worth as a formal structure.
4. The birth of Etech For Self-Worth marked the realization of the founders' aspirations to provide a platform for computer engineers to engage in impact projects while offering a range of services and training opportunities. The organization's incorporation under the relevant legal framework reflects its commitment to operating in compliance with the regulations governing associations in Cameroon.
5. Presentation of Etech:

#### Geographical Location:

O

***Figure 1 : Location plan of Etech for Self-Worth.***

#### Identification sheet:

##### ***Table 1 : Identification sheet of Etech For Self-Worth***

|  |  |  |  |
| --- | --- | --- | --- |
| **Headquarters** | Yaounde (ACCACIAS) | | |
| **Legal Form** | SARL | | |
| **Type of company** | Private | | |
| **Creation Date** | 15 October 2019 | | |
| **Services** | * Web Development * Infography * Digital Marketing * Mobile development * IOT * Electronics | | |
| **Chief Executive Office** | Mme ZANGUE Emilienne epse LALLAH | | |
| **Phone Contact** | (237) 673 55 22 37 | | |
| **Site Web** | www.etech-sw.org | | |
| **E-mail** | info@etech-sw.org |  |  |
| **Languages spoken** | French, English |  |  |

#### Hardware And Software Resources Of Etech

##### Hardware resources:

###### The company has many pieces of equipment listed in the following table.

###### ***Table 2 : Hardware resources of Etech For Self-Worth***

|  |  |
| --- | --- |
| 3 desktops BELINEA |  |
| 3 desktops HP L190T |  |
| 3 desktops SAMSUNG |  |
| 3 desktops MEDION |  |
| 3 desktops ACER |  |
| 1 printer RICOH |  |
| 1 printer canon IR2870 |  |
| 1 laptop HP |  |
| 1 laptop MAC |  |
| kit ultimate for raspberry |  |
| kit super starter |  |
| 2 drones |  |
| 2 smart robots |  |

##### Software resources:

###### ***Table 3: Software resources of Etech for Self-Worth***

|  |  |
| --- | --- |
| Système d’exploitation Windows 10 |  |
| Système d’exploitation Windows 8 |  |

1. **Visions and objectives:**

#### Visions:

Creating a self-sufficient generation of young people, our vision is to actively enable young people to reach their optimal potential. So that their maximum function is achieved, through the development of applicable skills for sustainable employment in society. All this for the development of Africa and Cameroon in particular.

#### Objectives:

Etech For Self-Worth has set the following objectives:

• Reduce poverty.

• Facilitate the development of human capacities.

• Holistic approach to young people (including educated youth, school dropouts).

• Support companies in their growth through techniques based on optimization, bottleneck management, and permanent profitability over time.

• Rights to development, security, and a better quality of life.

## Activities, Organizational Chart, and Responsibilities of Etech For Self-Worth:

#### Activities:

**Etech For Self-Worth** 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:

1. Mobile development
2. Web development
3. Internet of Things (IoT)
4. Artificial Intelligence
5. Data analysis
6. Big data
7. Machine Learning

#### Functional organization of Etech:

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.

**Etech For Self-Worth** has four divisions within it, and we have been assigned to all divisions except the secretariat. These are the divisions colored in yellow in the organizational chart below:

***Figure 2 : Organizational chart of Etech for self-worth***

#### Attributions:

##### General Management:

The General Management division is the largest decision-making unit within ETECH for Self-Worth. This division has the role of:

• Ensuring the smooth functioning of all departments of the company;

• Preparing the business plan and setting the general objectives of the company;

• Evaluating the realization of projects by the company.

##### 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.

##### 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.

##### IOT and Artificial intelligence Division:

In this division, all IOT, electronics and artificial intelligence projects of the company, partners and clients are routed for realization. Its role is to:

* Carry out electronic assemblies
* Study sensors and components
* Carry out market research
* Conduct training

##### Secretary:

##### For billing, photocopies, customer registration, and printing.

CONCLUSION

To conclude, our insertion in the environment was warm and convivial experience with the personnel and our professional supervisor. The most important thing i learned from this insertion phase is humility and communication. Our next phase here is the project specification which must be accurate and concise to fully grasp what is required to each other to bring fourth the best out of this live changing project.

# PART TWO: TECHNICAL PHASE

## 

|  |
| --- |
| BOOK I |
| EXISTING SITUATION |

Preamble

This section of our report will cover details of the different research did for the realization of this project. The existing study, criticism ot the existing system, problem statement, proposed solutions.

Content overview

INTRODUCTION

1. Theme/Project Presentation
2. Existing System Study
3. Critique of the Existing System
4. Problem Statement
5. Proposed Solution

CONCLUSION

INTRODUCTION

In this section of our report, we will delve into the various research conducted for the realization of this project. We will explore the existing study, critique the current system, identify the problem statement, and propose a solution. Through thorough analysis and investigation, we aim to provide a comprehensive understanding of the research undertaken to address the challenges and develop an effective solution. By examining the existing landscape and presenting our proposed approach, we strive to contribute towards the advancement and improvement of the project's objectives.

1. Theme/Project Presentation
2. Existing System Study

DESCRIPTION OF THE EXISTING SYSTEM

|  |  |  |
| --- | --- | --- |
| **LIMITATION CONSEQUENCE** | | **PROPOSED SOLUTION** |
| **The powering of the generator is done manually which may take time when in need of electricity during working ours or important event on campus** | People working at that moment or participating during the event may not get what is being said and waist of time is been observed | Our remote control and monitoring system will enable rapid powering of the generator and give statistics of its working hours |
| **Maintenance workers or guards can’t know amount of fuel in the generator** | When wanting to power it on it turns off after some time due to lack of fuel | Our remote control and monitoring system will enable workers in charge o know the fuel level of the generator in real time to know exactly if in need or not |
| **Workers in charge won’t know if there is abnormal voltage or frequency in the generator which can lead to certain malfunctioning at a point in time** | May over heat and damage some part of the generator | Our remote control and monitoring system will enable them to view at real time the amount of frequency in the generator and alarm in case of abnormality |
| **Workers in charge won’t know if there is abnormal voltage or frequency in the generator which can lead to certain malfunctioning at a point in time and could also lead to a disaster** | Disaster may happen which can either cost the generator or lives | Our remote control and monitoring system will enable E-stop which will cut off the system when ever it discovers abnormal functioning in it |
| **Some times workers may not know where to start whenever the generator get bad** | May need to open all the generator to analyze each part | Our remote control and monitoring system will enable Diagnostic codes to tell you where the fault occur and what can be done |

1. Critique of the Existing System
2. Problem Statement
3. Proposed Solution

CONCLUSION

In this section, we conducted research to support the realization of the Digital Smart Egg Incubator project.We analyzed the existing system, identified its limitations, and proposed a solution. Our findings contribute to improving the project's objectives. The current system relies on manual monitoring, lacking automation and precise control. This leads to sub optimal incubation conditions and reduced hatching success rates. The Digital Smart Egg Incubator integrates IoT technology, sensors, and a mobile application for real-time monitoring and control. It addresses the limitations of the existing system, enhancing efficiency and accuracy. Our research provides a foundation for the development of an advanced egg incubation system.

|  |
| --- |
| BOOK II |
| CONTEXT AND JUSTIFICATION |

Preamble

This section of our report will cover details on the product to be delivered with the agreement of the client(s) and the solution provider. In this section, we are going to present the context in which we are to but in place a platform, what the system should do and how the system should do it

Content overview

INTRODUCTION

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

CONCLUSION

## INTRODUCTION

The specification book gives us the different directives on the product to be delivered with the agreement of the client(s) and the solution provider. In this section, we are going to present the context in which we are to but in place a platform, what the system should do and how the system should do it. These specifications are to avoid the production of inadequate results. We will also include project detail such as the team involve, constraints, the budget, deadlines, constrains and the deliverable.

## CONTEXT AND JUSTIFICATION OF STUDIES

The development of the Digital Smart Egg Incubator project is driven by the need for a more efficient and reliable method of egg incubation. Traditional egg incubation methods often require manual monitoring and control, which can be labor-intensive and prone to human error. In order to address these limitations, the Digital Smart Egg Incubator project was initiated.

The purpose of the Digital Smart Egg Incubator is to utilize modern technology, specifically IoT (Internet of Things), to optimize the egg incubation process. By integrating sensors, micro controllers, and a mobile application, the project aims to automate and streamline the incubation process. This will result in improved accuracy, convenience, and overall success rates in hatching eggs.

The benefits of the Digital Smart Egg Incubator are multiform. Firstly, the use of IoT technology allows for real-time monitoring and control of environmental factors such as temperature, humidity, and ventilation. This ensures that the eggs are provided with the optimal conditions for successful incubation. Secondly, the mobile application provides users with remote access to monitor and manage the incubation process, offering convenience and flexibility. Users can receive notifications, adjust settings, and track the progress of the eggs, all from their mobile devices.

Furthermore, the Digital Smart Egg Incubator has the potential to benefit various industries and individuals. It can be utilized by poultry farmers, hatcheries, and researchers involved in egg incubation. Additionally, it can be a valuable tool for educational institutions teaching poultry science and biology, allowing students to gain hands-on experience in a technologically advanced incubation process.

Overall, the development of the Digital Smart Egg Incubator is justified by the need for a more efficient, accurate, and user-friendly approach to egg incubation. By leveraging IoT technology and a mobile application, this project aims to revolutionize the incubation process, leading to improved outcomes and increased productivity in the field of egg hatching.

## PROBLEM DEFINITION

The project aims to enhance traditional egg incubation methods by addressing several key challenges using IoT technology.

One challenge is the manual monitoring and control required in traditional incubation methods. This process is time-consuming, prone to errors, and inconvenient for users. To overcome these challenges, the project focuses on automating the incubation process using IoT technology, eliminating the need for constant manual intervention.

Maintaining optimal environmental conditions is crucial for successful egg incubation. Traditional incubators often lack precise mechanisms for regulating temperature and humidity levels. To overcome this limitation, the project integrates sensors, a micro-controller, and relay modules to automate temperature and humidity regulation, ensuring optimal conditions for higher hatching success rates.

The project aims to enhance traditional egg incubation methods by addressing several key challenges using IoT technology.

One challenge is the manual monitoring and control required in traditional incubation methods. This process is time-consuming, prone to errors, and inconvenient for users. To overcome these challenges, the project focuses on automating the incubation process using IoT technology, eliminating the need for constant manual intervention.

Maintaining optimal environmental conditions is crucial for successful egg incubation. Traditional incubators often lack precise mechanisms for regulating temperature and humidity levels. To overcome this limitation, the project integrates sensors, a micro-controller, and relay modules to automate temperature and humidity regulation, ensuring optimal conditions for higher hatching success rates.

The presence of ammonia gas can indicate the presence of bad eggs, which negatively impacts the incubation process. To address this issue, a gas sensor (MQ-135) is incorporated into the system. This sensor detects ammonia gas levels within the incubator, enabling timely actions to maintain a healthy incubation environment.

Unauthorized movement or disturbance within the incubator can disrupt the incubation process. To mitigate this problem, the project includes a motion sensor (PIR sensor). This sensor detects any movement within the incubator and alerts users, allowing them to identify and address potential disturbances promptly.

To enhance user convenience, a mobile application is developed. The app serves as a user-friendly interface, enabling remote monitoring and control of the incubation process. Users can access real-time data, adjust settings, receive notifications about critical events, and track the progress of each incubation cycle conveniently through the mobile application.

By addressing these challenges, the project aims to improve the efficiency, accuracy, and convenience of egg incubation. The integration of IoT technology, including automated control, environmental regulation, gas detection, movement detection, and the mobile application, enhances the success rates and user experience throughout the incubation journey.

## OBJECTIVES

A. General Objective

The general objective in this project is to develop an IoT-based egg incubator along with a mobile application that provides automated control and monitoring of the incubation process. The goal is to optimize the hatching process and improve the overall success rate of egg incubation.

1. General Objective

Administrator:

* Access the back-end administration panel to manage user accounts, including creating, modifying, or deleting accounts for farmers and visitors.
* Monitor the overall system performance, including data transmission, sensor readings, and connectivity status.
* Receive notifications or alerts regarding any critical issues or malfunctions within the incubator system.
* View and analyze data collected from different incubators, such as hatch success rates, temperature profiles, and humidity trends.
* Generate comprehensive reports and analysis to assess the performance of the incubator system and identify areas for improvement.
* Manage User’s accounts (enable, disable, modify) and generate incubator’s CODE.
* Provide technical support and guidance to farmers or visitors facing any issues or difficulties with the app or incubator system.

Farmer:

* Log in to the mobile application using their account credentials.
* Access a dashboard displaying all registered incubators and their corresponding status (e.g., active, paused, completed).
* Monitor real-time data from each incubator, including temperature, humidity, gas levels, water levels, and inclination.
* Receive push notifications and alerts for critical events, such as temperature or humidity deviations, gas detection, or completion of an incubation cycle.
* Adjust incubation settings remotely, such as temperature and humidity thresholds, to ensure optimal conditions for successful hatching.
* Track and analyze incubation results for each incubator, including the number of eggs initially placed, the number of good eggs, and the number of bad eggs.
* Access historical data and trends to evaluate the performance of past incubation cycles.
* Communicate with the administrator through in-app messaging for any technical support or inquiries.

Visitor:

* Download and install the mobile application from an app store.
* Explore general information about the automated egg incubator system, its features, and benefits.
* Access educational resources or guides related to egg incubation and farming practices.
* View a demo or simulated version of the app to understand its functionalities.
* Contact the administrator or request additional information through the platform's contact channels, such as email or a feedback form.

These functionalities allow the administrator to oversee the overall system, provide support, and analyze data. Farmers can remotely monitor and control multiple incubators, track incubation progress, and receive notifications. Visitors have access to information and resources, as well as channels to communicate with the administrator for further assistance.

## EXPRESSION OF NEEDS

1. Functional Needs

**Administrator:**

* User Management:
  + - 1. Register and manage user accounts, including authentication and access privileges.
      2. Handle user-related issues, such as password resets and account verifications.
* Incubator Management:
  + - 1. Add and manage incubators within the application.
      2. Assign incubators to farmers and monitor their status.
* Sensor Monitoring and Alerts:
  + - 1. Monitor real-time sensor data from incubators, including temperature, humidity, gas levels, and movement sensing.
      2. Receive notifications and alerts for critical events, such as deviations in temperature or humidity, gas detection, or system malfunctions.
* Data Analysis and Reporting:
  + - 1. Analyze incubation data, such as hatch success rates, temperature profiles, and humidity trends.
      2. Generate reports and insights on incubation performance for further analysis and decision-making.
* System Maintenance and Updates:
  + - 1. Perform maintenance tasks on the incubator system, including firmware updates and sensor calibrations.
      2. Ensure the system is up-to-date and functioning properly.

**Farmer:**

* Incubator Monitoring:
  + - 1. Monitor real-time sensor data from assigned incubators, including temperature, humidity, gas levels, and movement sensing.
      2. Receive notifications and alerts for critical events related to their specific incubators.
* Incubation Settings:
  + - 1. Adjust incubation settings remotely, such as temperature and humidity thresholds, to maintain optimal conditions for successful hatching.
      2. Set up automated schedules for specific incubation periods.
* Data Analysis and Tracking:
  + - 1. Track and analyze incubation results for each incubator, including the number of eggs initially placed, the number of good eggs, and the number of bad eggs.
      2. Access historical data and trends to evaluate the performance of past incubation cycles.
* Communication with Administrator:
  + - 1. Communicate with the administrator through in-app messaging for technical support or inquiries related to the incubator system.
      2. Provide feedback or report issues related to the incubator system.

**Visitor:**

* Application Information:
  + - 1. Access general information about the automated egg incubator system, its features, and benefits.
      2. Learn about the incubation process and outcomes achieved by registered farmers.
* Contact Administrator:

1. Contact the administrator or request additional information through the platform's contact channels, such as email or a feedback form.
2. Seek clarification or assistance regarding the automated egg incubator system.
3. Non-Functional Needs

* **Performance:**

1. Responsiveness: The application should respond quickly to user interactions and provide a smooth user experience.
2. Efficiency: The application should be optimized to perform tasks efficiently, minimizing resource usage and response times.
3. Scalability: The application should be capable of handling increasing user loads and data volumes without significant performance degradation.
4. Reliability: The application should be reliable and available, minimizing downtime and disruptions.

* **Usability:**
  + 1. Intuitive Interface: The application should have a user-friendly and intuitive interface that is easy to navigate and understand.
    2. Accessibility: The application should be accessible to users with disabilities, complying with accessibility standards and guidelines.
    3. Multilingual Support: The application should support multiple languages to cater to a diverse user base.
    4. Consistency: The application should maintain consistent design elements and interactions throughout different screens and modules.
* **Security:**

1. User Data Protection: The application should securely store and handle user data, following best practices for encryption and data privacy.
2. Authentication and Authorization: The application should provide secure user authentication mechanisms and access controls to protect user accounts and data.
3. Secure Transactions: The application should ensure secure transmission and handling of sensitive information, such as payment details.
4. Data Backup and Recovery: The application should have robust backup and recovery mechanisms to protect against data loss and enable quick recovery.

* **Compatibility:**

1. Cross-Platform Compatibility: The application should be compatible with different operating systems and devices, including mobile and web platforms.
2. Browser Compatibility: The web application should be compatible with major web browsers, ensuring consistent functionality and appearance.
3. API Integration: The application should be able to integrate with external systems or APIs seamlessly.

* **Performance Monitoring and Analysis:**

1. Logging and Monitoring: The application should log events and errors for monitoring and troubleshooting purposes.
2. Analytics and Insights: The application should provide analytics and insights on user behavior, system performance, and usage patterns.
3. Performance Optimization: The application should continuously optimize its performance based on analytics and monitoring data.

* **Maintainability:**

1. Modularity: The application should be designed with a modular structure, making it easier to maintain and enhance individual components.
2. Code Quality: The application's code should follow best practices and coding standards, ensuring readability and maintainability.
3. Documentation: The application should have comprehensive documentation, including installation instructions, user guides, and API documentation.

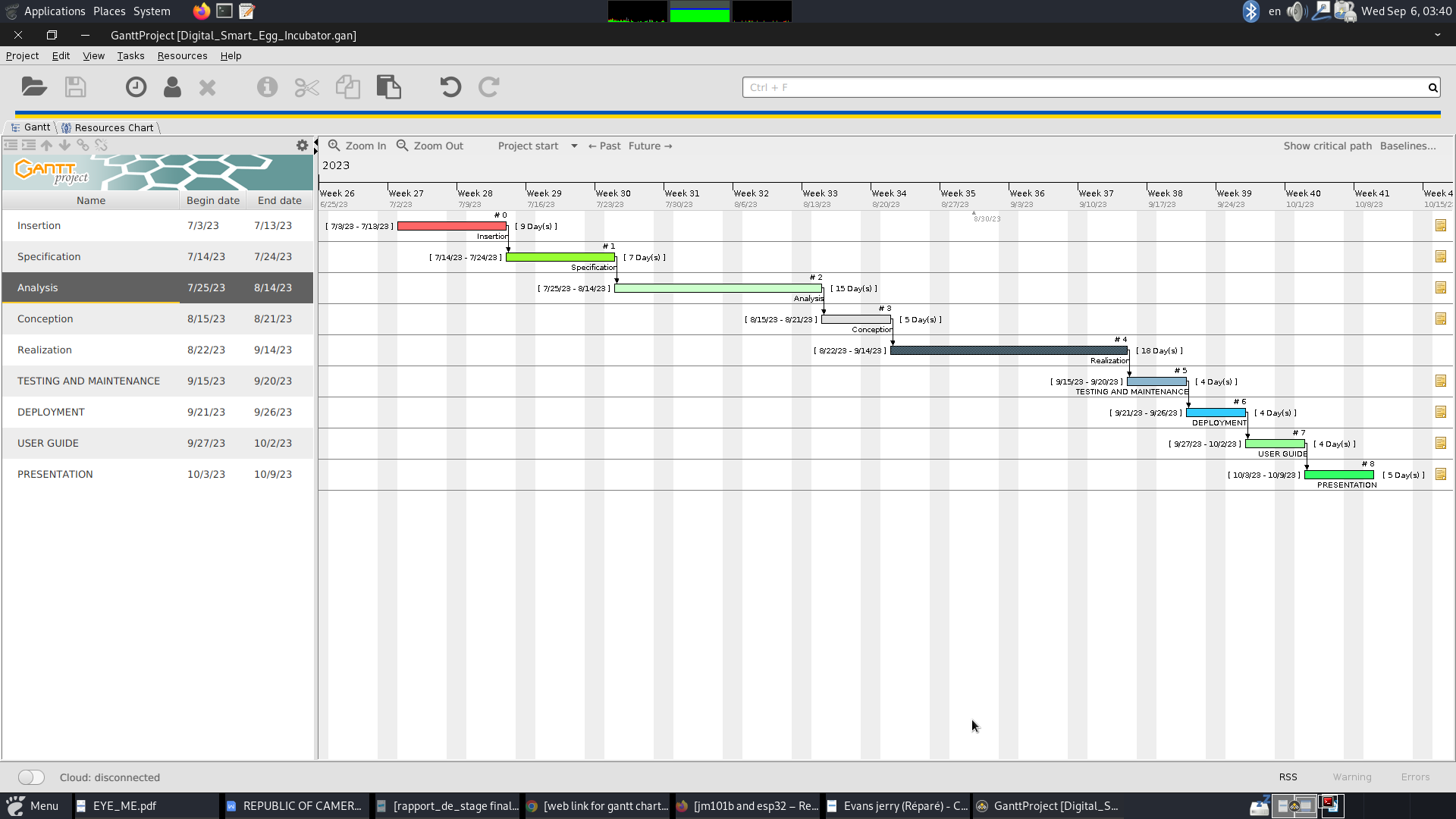
* **Integration and Interoperability:**

1. Integration Capabilities: The application should support integration with third-party services, platforms, or APIs to extend its functionality
2. Interoperability: The application should be able to exchange data and interact with other systems effectively.Human resources are vital for developing a digital smart egg incubator using IoT technology. Engineers in IoT and embedded systems design the hardware and software components. Poultry farming and biology experts optimize the incubation process. Project managers coordinate tasks and ensure timely delivery. Collaboration and teamwork among these professionals are crucial. The goal is to create an incubator that maximizes hatching success and sup-human resources are vital for developing a digital smart egg incubator using IoT technology. Engineers in IoT and embedded systems design the hardware and software components. Poultry farming and biology experts optimize the incubation process. Project managers coordinate tasks and ensure timely delivery. Collaboration and teamwork among these professionals are crucial. The goal is to create an incubator that maximizes hatching success and supports sustainable poultry production.ports sustainable poultry production.

## PLANNING THE PROJECT

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Objective** | **Outing** | **Duration** |
| INSERTION | This stage involves gathering all the relevant information and requirements for the project. | Insertion report | 10 days |
| SPECIFICATION BOOK | The specification book is a document that outlines the detailed specifications and requirements of the automated egg incubator system. It includes the functional and non-functional requirements. | Specification book | 10 days |
| ANALYSIS | The analysis stage involves analyzing the gathered requirements and conducting a feasibility study. This helps determine the technical and economic viability of the project. | Analysis | 20 days |
| CONCEPTION | In the conception stage, the project's conceptual design is developed. This includes defining the system architecture, selecting the appropriate technologies, and creating the initial design project plan | Conception document | 5 days |
| REALIZATION | Build the automated egg incubator system by designing and assembling hardware components, developing software, integrating sensors and controls, and ensuring proper functionality. | Realization Document | 25 days |
| TESTING AND MAINTENANCE | Thoroughly test the system to verify that it meets specified requirements, perform maintenance activities like bug fixes and updates, and ensure ongoing system performance. | User testing | 5 days |
| DEPLOYMENT | Install and configure the automated egg incubator system in the intended environment, making it ready for use and ensuring compatibility with user requirements. | Development | 5 days |
| USER GUIDE | Provide comprehensive documentation and instructions on system operation, user interface navigation, troubleshooting, and maintenance procedures. | User guide | 5 days |
| PRESENTATION | Showcase the completed project to stakeholders through demonstrations, highlighting its functionality, features, and benefits. | PowerPoint | 5 days |

Gantt Diagram



1. Actors of The Project

|  |  |  |
| --- | --- | --- |
| Name | Function | Role |
| Mr. MESSIO Ulrich | Lecturer at the African Institute of  Computer Sciences. | Academic Supervisor |
| Mr. LALLAH Raoul | DevOps | Professional Supervisor |
| Mr. GUIFFUO Joel | * Designing and Developing Electronic Systems. * Testing and Troubleshooting. * Prototyping and Product Development. * Circuit Design and PCB Layout. | Electronics Engineer |
| Mr. TAYOUTSOP KANOU Edson Rael | * Project Planning * Resource Allocation * Team Management * Risk Management | IoT Project Manager |
| ESSONO Jordan Ryan | Student at the African Institute of  Computer Sciences. | Project head, analyst, conception and  coding of the project , testing |
|  |  |  |

## ESTIMATE OF THE PROJECT

A. Software Resources

|  |  |  |  |
| --- | --- | --- | --- |
| Software | Usage | Quantity | Price(FCFA) |
| WPS Office | Used for the creation of the report and the  PowerPoint. | 1 | 287,500 |
| Google chrome | A free and open source web browser use to make  research on the project. | 1 | free |
| Parrot Security System | A system software uses to manage our computer  hardware and software. | 1 | free |
| WSGI Server | A local server use to create and manage our  database. | 1 | free |
| GANTT Project | Used for the planning of the different task that are  necessary for the realization of our project. | 1 | free |
| Visual Paradigm | Modeling tool which was use for the modeling of  the system to be developed.REALISATION PHASE | 1 | free |
| Visual studio Code | Front-End Development environment | 1 | free |
| Postman | API development and testing tool | 1 | free |
| Pycharm Community | Back-end Development | 1 | free |
| Total 1 | | | 313,500FCFA |

1. Hardware Resources

Source: mercurial

|  |  |  |  |
| --- | --- | --- | --- |
| Software | Usage | Quantity | Price(FCFA) |
| Computer(Dell Latitude E7450) | Used for the creation of the report and the  PowerPoint. | 1 | 287,500 |
| PrinterHP officejet  4630 | A free and open source web browser use to make  research on the project. | 1 | free |
| Modem Jio | A system software uses to manage our computer  hardware and software. | 1 | free |
| WSGI Server | A local server use to create and manage our  database. | 1 | free |
| GANTT Project | Used for the planning of the different task that are  necessary for the realization of our project. | 1 | free |
| Visual Paradigm | Modeling tool which was use for the modeling of  the system to be developed. | 1 | free |
| Visual studio | Front-End Development environment | 1 | free |
| Postman | API development and testing tool | 1 | free |
| Pycharm Community | Back-end Development | 1 | free |
| Total 1 | | | 313,500FCFA |

1. Human Resources

Human resources are vital for developing a digital smart egg incubator using IoT technology. Engineers in IoT and embedded systems design the hardware and software components. Poultry farming and biology experts optimize the incubation process. Project managers coordinate tasks and ensure timely delivery. Collaboration and teamwork among these professionals are crucial. The goal is to create an incubator that maximizes hatching success and supports sustainable poultry production.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RESOURCES |  | NUMBER | COST PER DAY | NUMBER OF DAYS | COST (FCFA) |
| Project manager | 1 |  | 250000 | 60 | 15000000 |
| Analyst | 1 |  | 150000 | 30 | 4500000 |
| Designer | 1 |  | 130000 | 25 | 3250000 |
| Programmer | 1 |  | 100000 | 32 | 3200000 |
| TOTAL 3 | 4 |  |  |  | 25950000 |

## CONSTRAINTS

## DELIVRABLES

At the end of the specified development timeline for our software, the deliverable will consist of a Report that will include:

* A requirements document,
* An analysis dossier,
* A design dossier ,
* An implementation dossier,
* The source code of the applications,
* A user guide.

## CONCLUSION

The specifications book contains detailed information about the project's objectives, requirements (both functional and non-functional), team members, project plan, cost, and constraints. This document serves as a guide to develop a solution that meets the project's needs while avoiding errors and compatibility issues. The specifications book acts as a foundation for the analysis phase of the project, where we carefully examine the provided information to determine the best approach and ensure a successful outcome.

|  |
| --- |
| BOOK III |
| 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.

Content overview

INTRODUCTION

1. Methodology
2. Comparative Study of UML and MERISE
3. Comparative Study of Unified Process
4. Modeling
5. Use Case Diagram
6. Communication Diagram
7. Sequence Diagram
8. Activity Diagram

CONCLUSION

## INTRODUCTION

The analysis book permits us to examine in an explicit way the existing system, it’s 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.

## PRESENTATION OF ANALYSIS METHOD

1. Comparative Study of UML and MERISE

We have studied in details some analysis methods by looking at its objectives, its structuring, its pros and cons, so as to choose an analysis method which is reliable and adaptive to our project, below are some analysis methods we studied and a brief explanation in order for you to understand why we made our choice;

* 1. **MERISE:** it is an information system design and development widely used in France. It was first early introduced in 1980s.
  2. **Agile:** this methodology is growing in popularity, thanks to highly competitive business environment and increased innovation. In general, agile methodologies prioritized shorter, interactive cycle and flexibility.
  3. **Scrum:** this is the most popular agile development framework because it is relatively simple to implement. It also solves so many problems that software developers struggle with in the past, convoluted development cycles, project plan, and shifting production schedules. This methodology allows for rapid development and testing, especially with small teams.
  4. **APF:**  which stands for ADAPTIVE PROJECT FRAMEWORK, it grows from the difficulty in managing most IT projects using traditional project management methods due to uncertain and changing requirement. APF begins with a requirement breakdown structure (RBS) to define strategic goals based on productive requirements, functions, sub-function and features. The project proceeds in iterative stages, and at the end of each step, teams evaluate previous results to improve performance and practices.

**XP:** which stands for Extreme Programming is a software development methodology that advocate frequent releases in short development cycles, which is intended to include checkpoints for the adoption of new customer requirements and improve productivity. This methodology takes its name from the idea that the traditional software engineering practices are taken to extreme levels.

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 it 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

1. 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.

1. 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 :

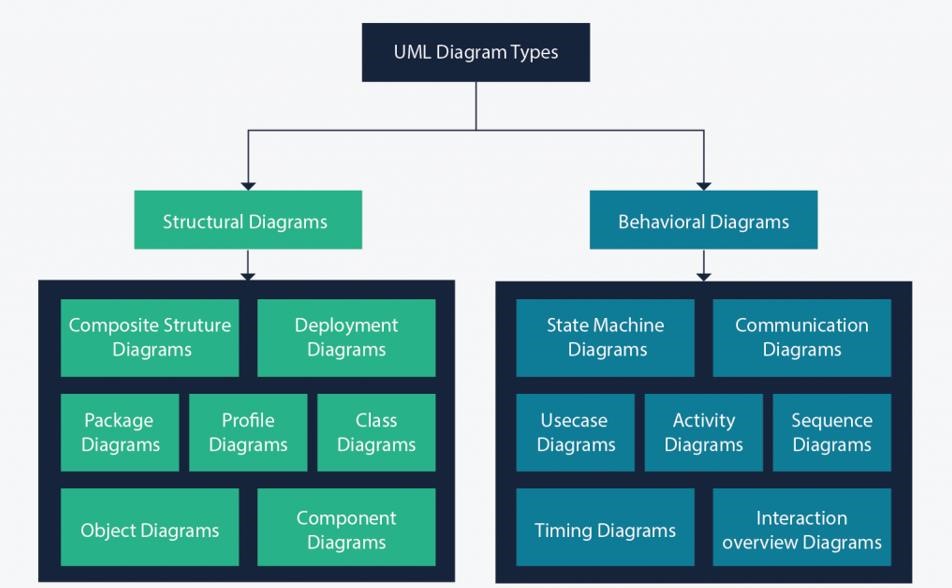


Figure 4: UML 2.5 diagrams overview (source: https://creately.com/blog/diagrams/uml-diagram-types-examples/)

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.

1. Comparative Study of Unified Process
2. The Process of Development of a Software

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.

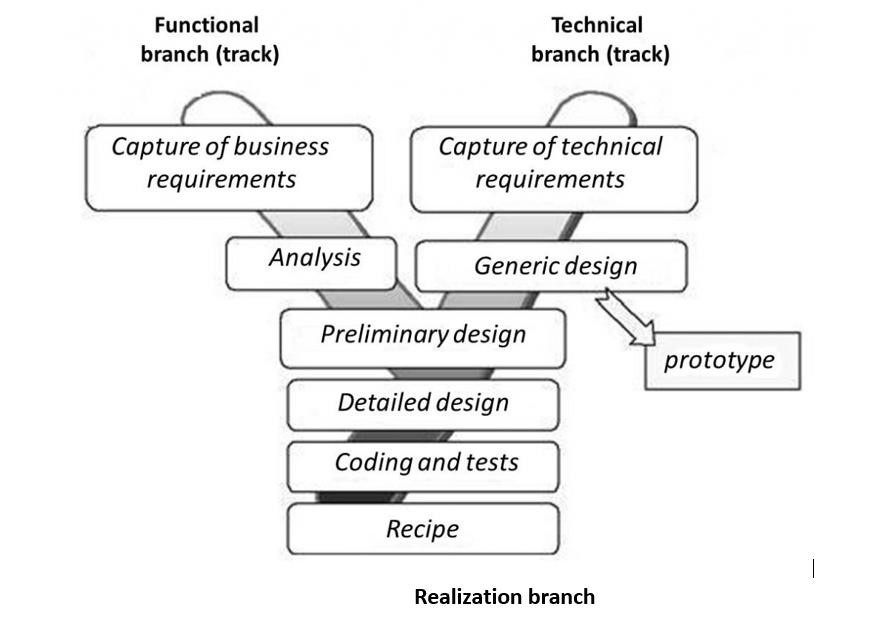


Figure 5: 2TUP diagram (Source: [https://www.mysciencework.com/omniscience/pervasive-mobile-healthcare-system-basedon-cloud-computing](https://www.mysciencework.com/omniscience/pervasive-mobile-healthcare-system-based-on-cloud-computing)*[)](https://www.mysciencework.com/omniscience/pervasive-mobile-healthcare-system-based-on-cloud-computing)*

##### 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, inter-phases 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 deliverable 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 visualize 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 visualize 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 work-flow 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 SYSTEM

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.

#### Use Case Diagram:

##### Definition:

A use case diagram shows the functionalities of a system, their inter-dependencies 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:

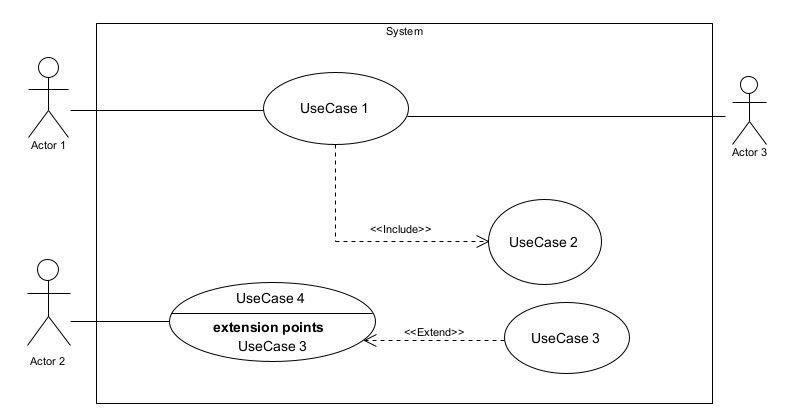


Figure 6: Use Case diagram formalism

Table 10 :Use case diagram component

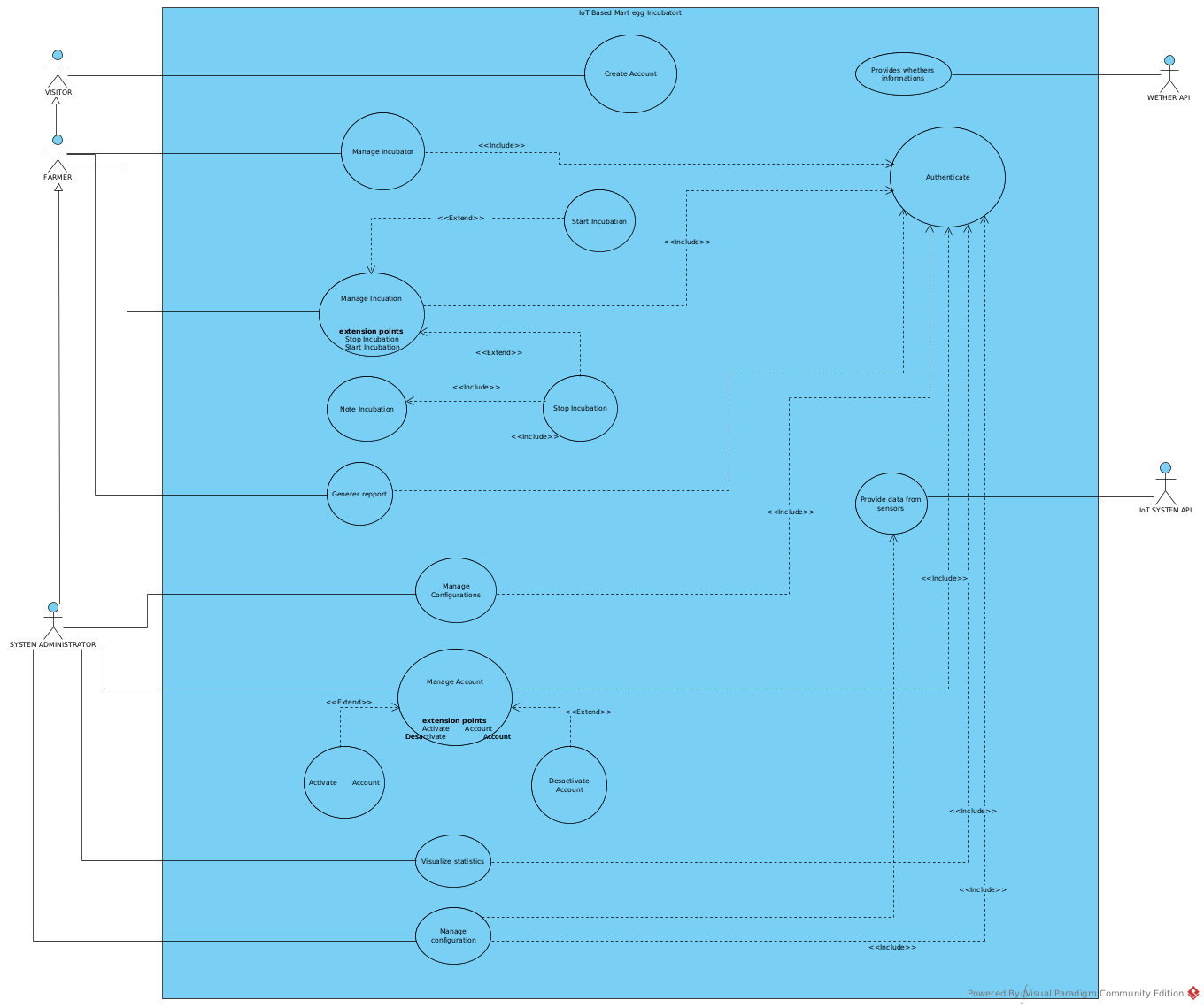
|  |  |  |
| --- | --- | --- |
| Elements Notation Description | | |
| Actors |  | Represents an entity that directly interacts with the system. The actor is what performs the different possible actions of the system. |
| Use case |  | A use case represents a |
|  | Use Case  1 | functionality of the 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 11:Actors of our system.*

|  |  |
| --- | --- |
| **Actor** | **Role** |
| **Visitor** | Their responsibility is to follow-up dear children at real-time. They can view attendance, school fee status, and other vital information. |
| **User** | He is in charge of recording attendance of students, on daily basis. |
| **System Administrator** | He can identify a random student at the campus, during exams or on special occasions. |

##### General Use Case Diagram



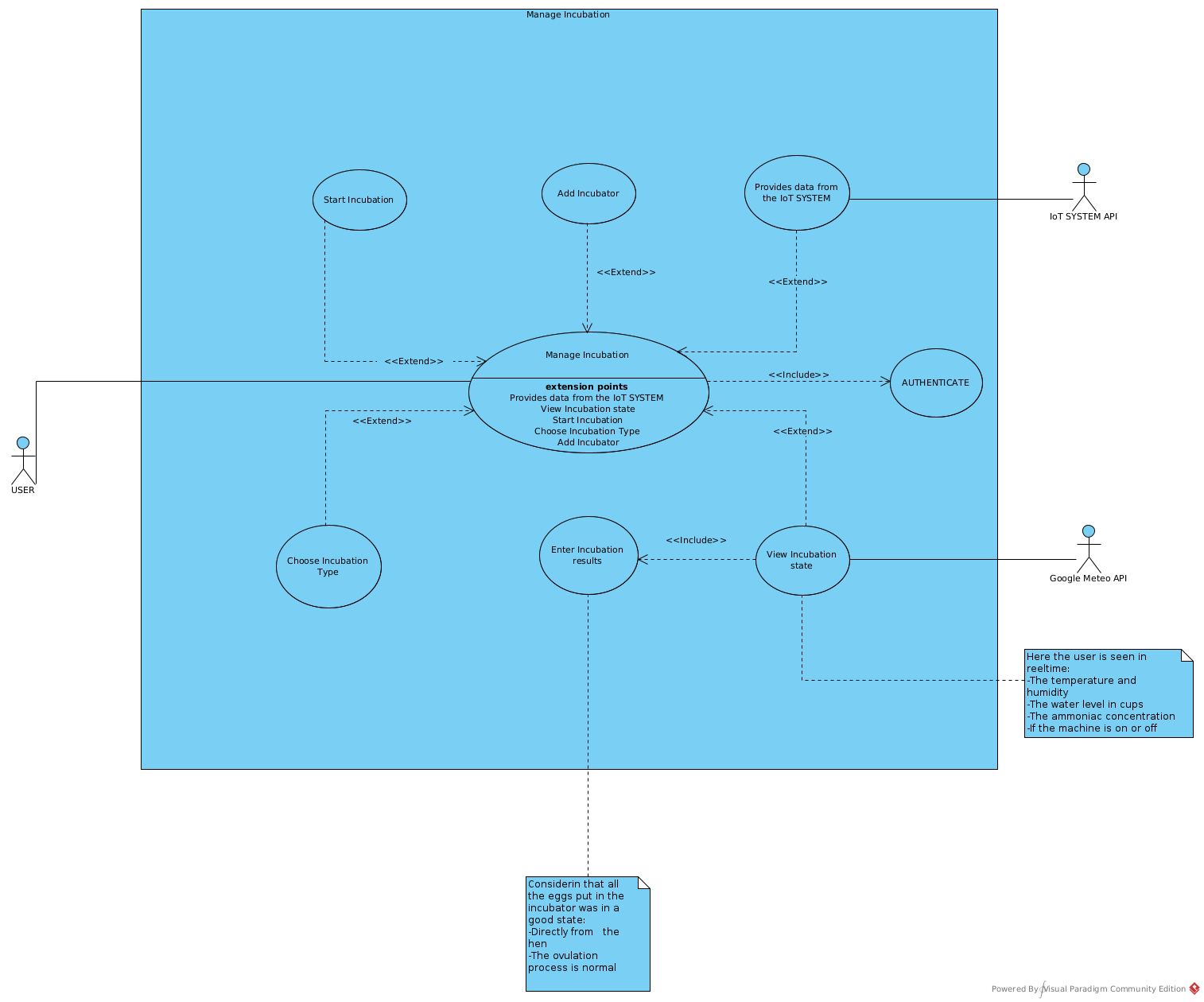
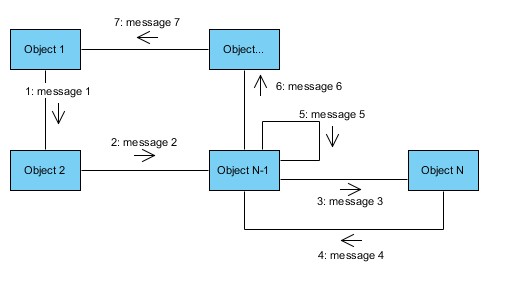


Figure 7 :General Use Case Diagram

1. **Communication Diagram**
   1. Definitions:

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.

* 1. Formalism



*Figure 29:Communication Diagram Formalism*

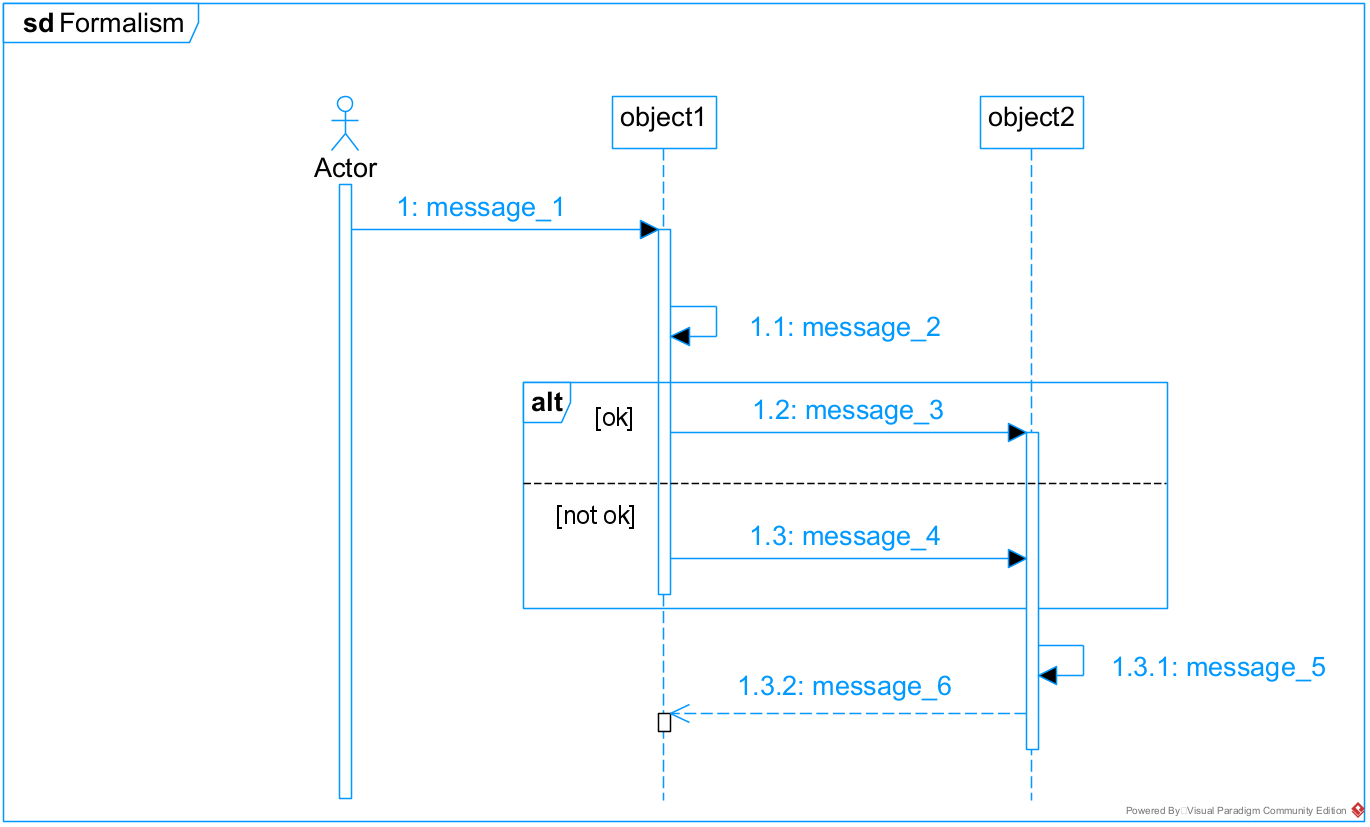
*Table 22:Communication Diagram Components*

|  |  |  |
| --- | --- | --- |
| **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. |

1. **Sequence Diagram:**
2. Definition:

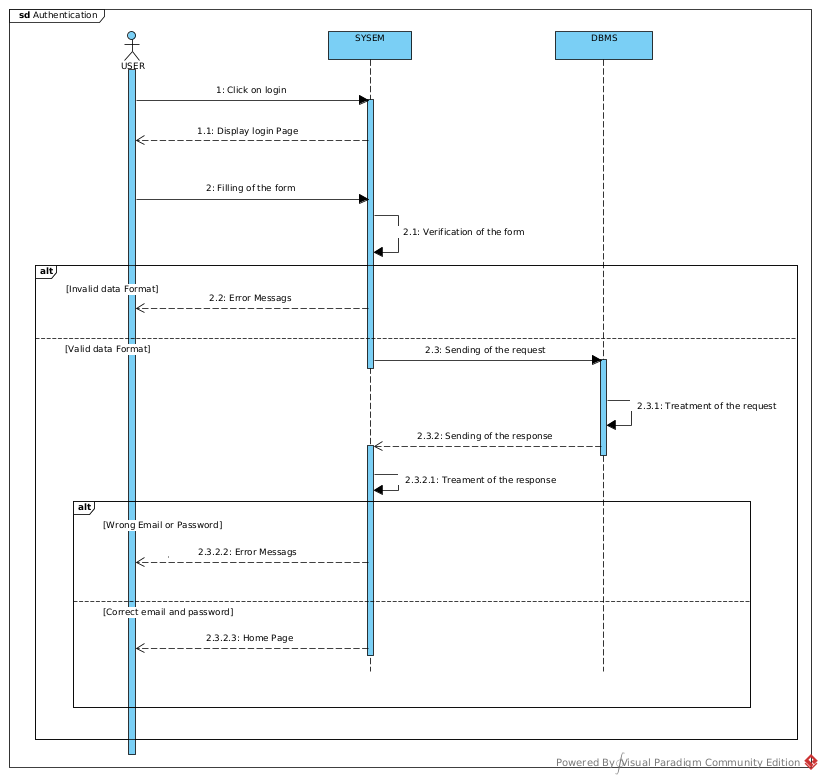
A sequence diagram is an interaction diagram which represents the flow of message between elements in a system, it is termed as an event diagram. It portrays the communication between any two lifelines as a time-ordered sequence of events.

1. Formalism:



*Table 19:Sequence Diagram Components*

|  |  |  |
| --- | --- | --- |
| **Element** | **Representation** | **Description** |
| **Lifeline** |  | An individual participant in a sequence diagram, it is position at the top of the diagram. |
| **Combined fragment** |  | It represents a choice of behavior in which at most one operand will be chosen. |
| **Messages** |  | These are arrows which shows the direction of message flow. We have the synchronous, the asynchronous and the selfmessages. |
| **Activation** |  | It describes the time period in which an operation is performed by an element. |



#### Activity Diagram

##### Definition:

An activity diagram is a graphical representation of work-flows 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

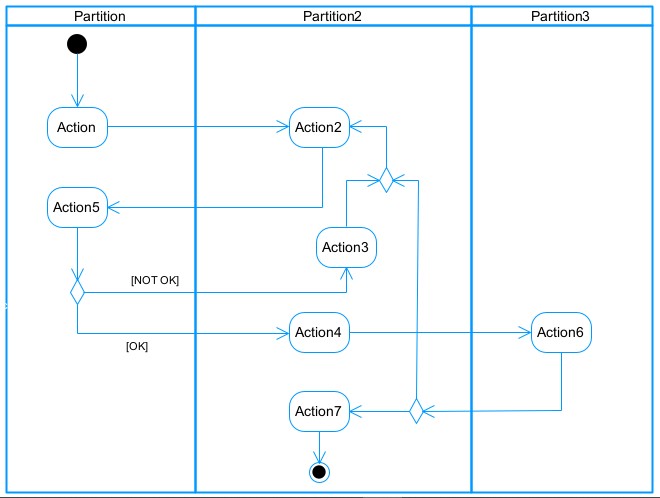
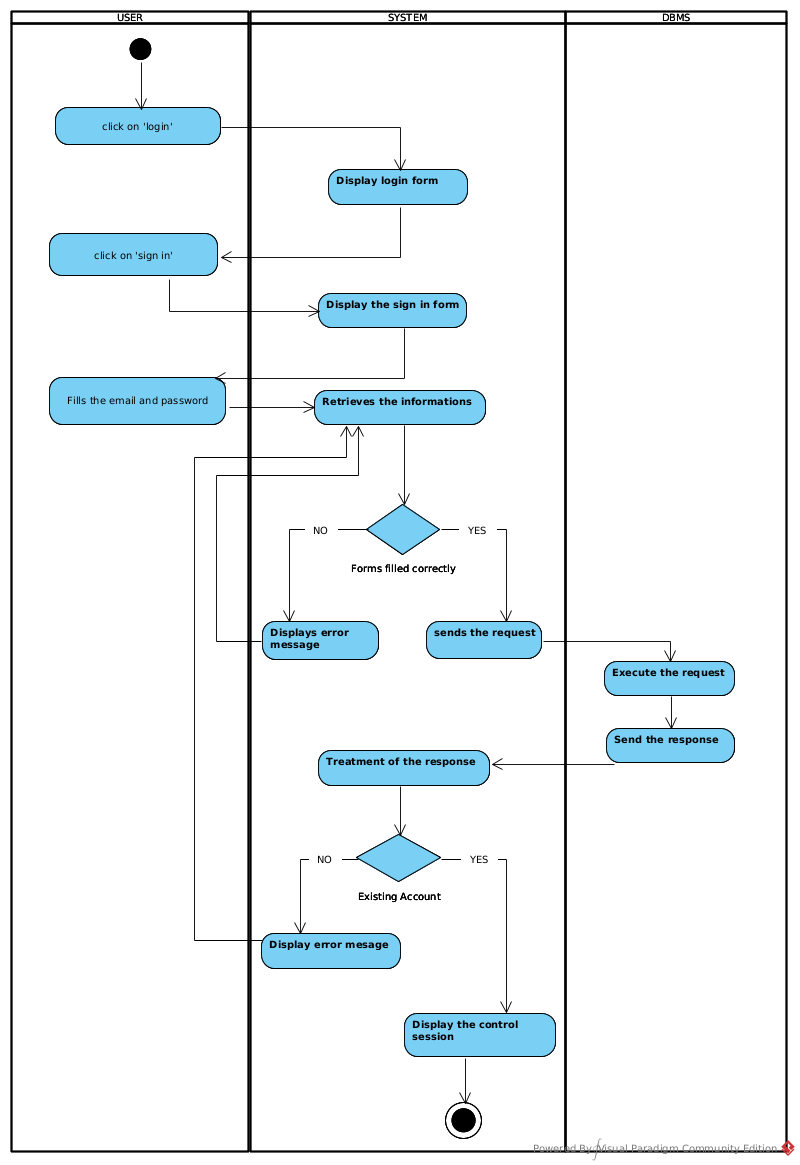
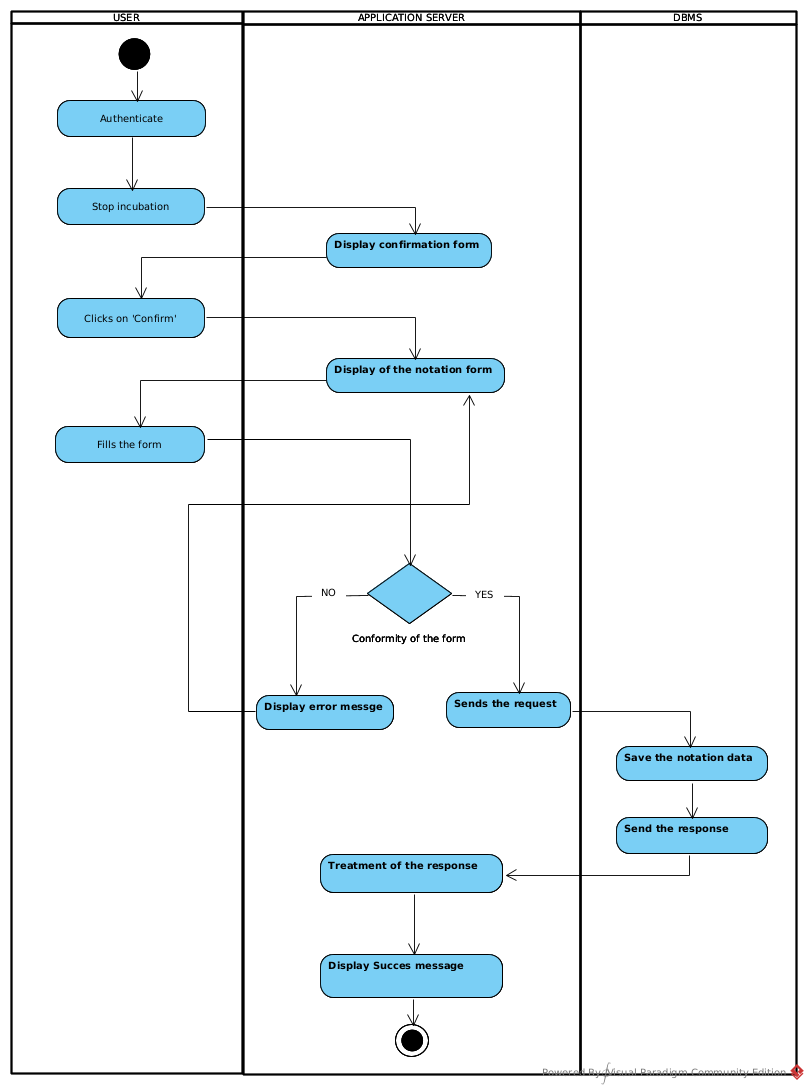


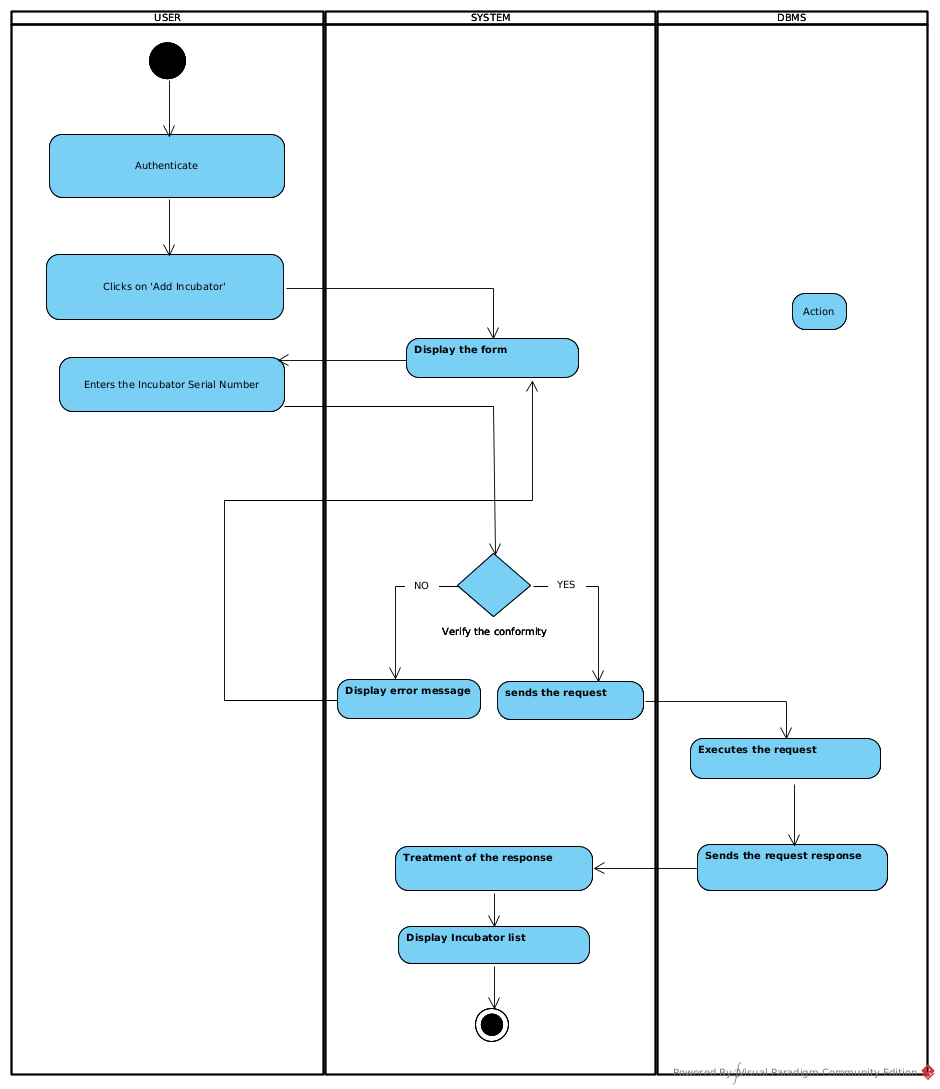
Figure : Activity Diagram Formalism

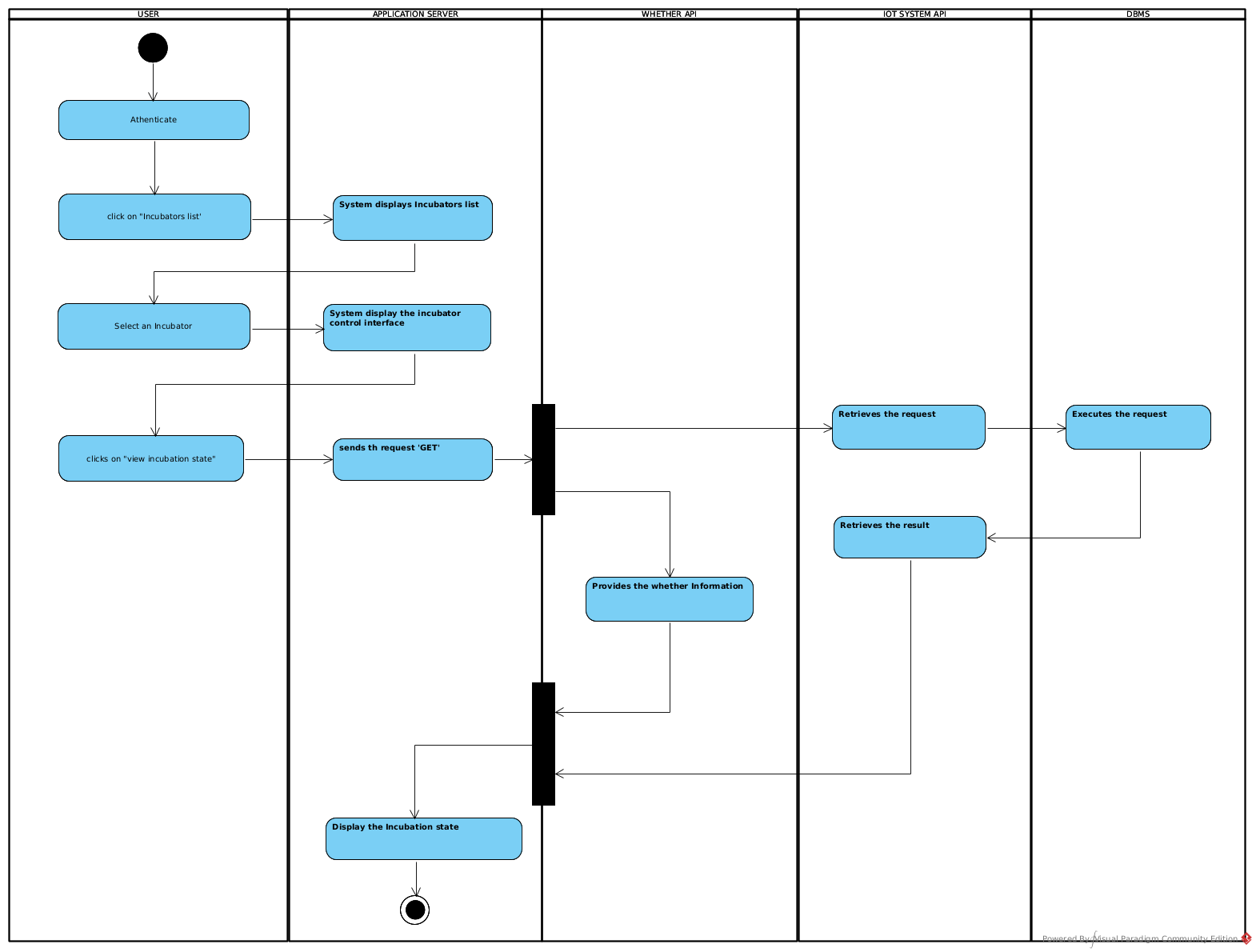
Table :components of an Activity Diagram

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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. | |







CONCLUSION

In the analysis phase, we chose a software development process and modelling language, after which we presented the existing system, its limitation and our proposed solutions. We went forth explaining the functional need of our system, we saw the use case diagram which shows the relationship between the actors and use case (the action the actor can perform on the system), we saw the activity diagram which shows the work-flow of our system, and lastly the State machine diagram which shows the behavior of a single object in response to an event. We will now move to the conception phase in which we will present the other two branches of the 2TUP which are the Technical and the realization branch of our system.

|  |
| --- |
| BOOK 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 overview

INTRODUCTION

1. Class Diagram
2. State Transition Diagram
3. Package Diagram

CONCLUSION

## INTRODUCTIOCN

The conceptual phase will describe in details the necessary specifications, features and operations that will satisfy the functioning requirements of the proposed system as modeled in the analysis phase. This phase is meant to identify and consider essential components (hardware /or software), structure (network capabilities), processes and procedures for the system to accomplish it objectives. We will look at some diagrams such as the component diagram, package diagram, communication diagram, class diagram and the object diagram.

1. Content Overview **Class Diagram:**
   1. 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 also constraints imposed on the system. It purpose is to model the static view of an application

* 1. Formalism:

multiplicity

private (accessible

only within the class

scope)

public

(

accessible out

of the class

scope)

0..1

0..\*



class name

-

Attribute

:

int



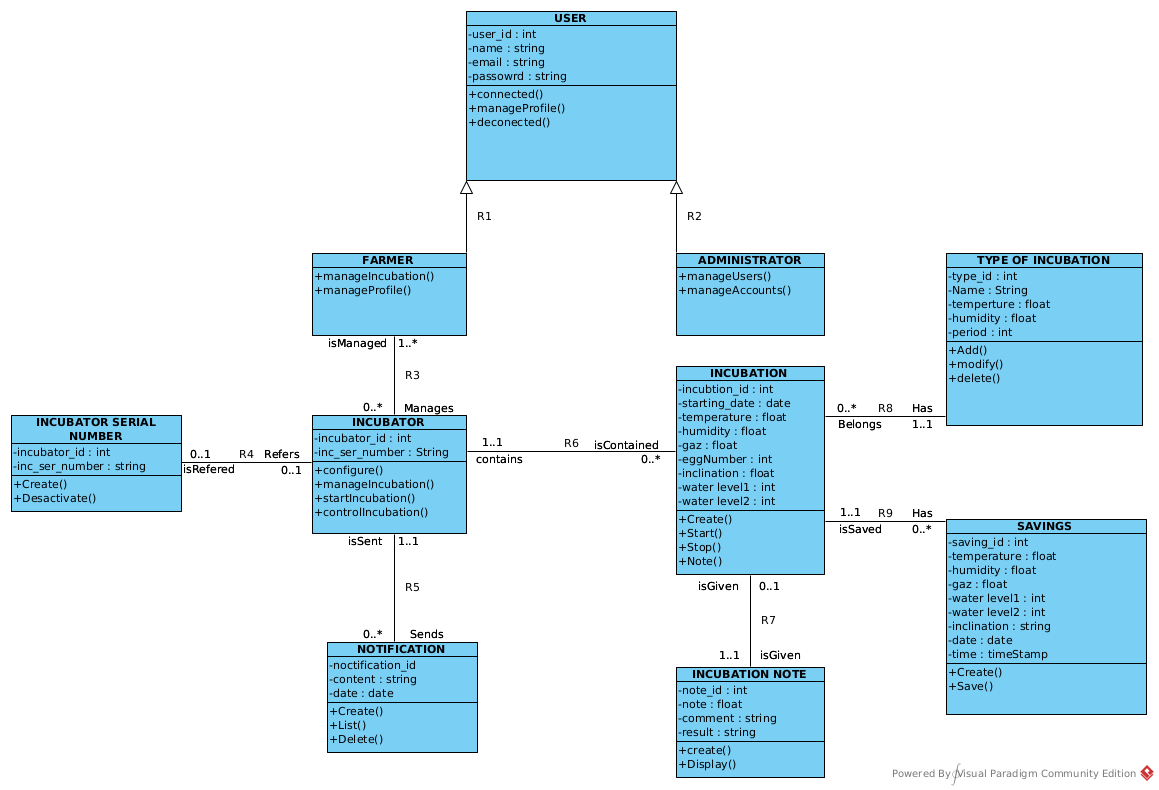
Classe\_2

*Figure 34:Class Diagram Formalism*

*Table 24:Class Diagram Components*

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

* 1. System Class Diagram:



* 1. Relationship between Classes

**R1:**

**R2:**

**R3:**

**R4:**

**R5:**

**R6:**

**R7:**

**R8:**

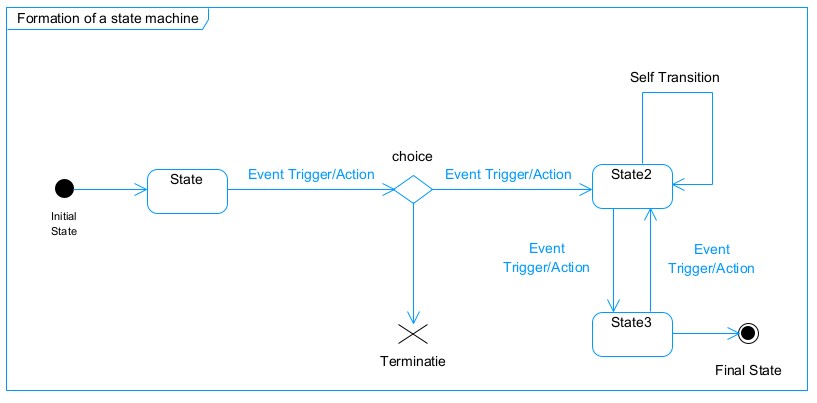
**R9:State Machine Diagram**

##### 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.

##### Formalism:

*Figure :Formalism of a state machine diagram*



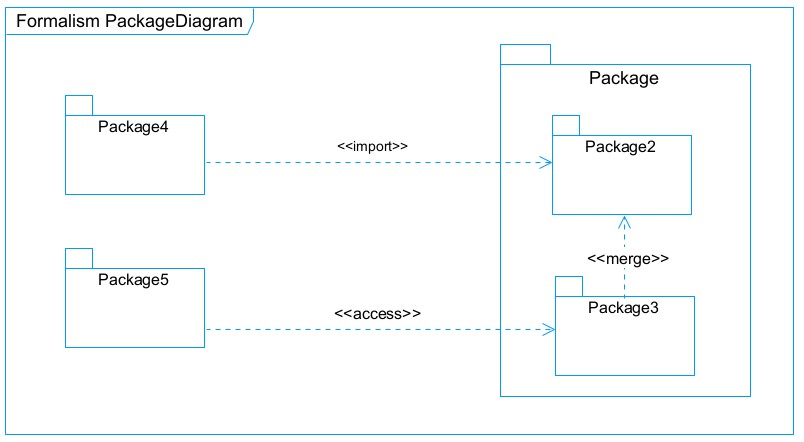
*Table 16 State Machine Diagram Components:*

|  |  |  |
| --- | --- | --- |
|  | **Diagram Relationship** | **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 it context is terminated. |
| **Diagram Overview** |  | A placeholder for the linked states in a state machine diagram. |

1. **Package Diagram:**
   1. 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 eve other packages.

* 1. Formalism:



*Figure 21:Package Diagram Formalism*

|  |  |  |
| --- | --- | --- |
| **NAME** | **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. |

*Figure 22:Package Diagram Components*

## CONCLUSION

In the conception phase, we set as objective to plane the different aspect of our system by showing how it will be structure and deployed within existing technical architectures. We began by taking into account the technical constraints for our system, after which we proceeded to identify the components of our system, how they are grouped together and how they should be deployed on appropriate deployment targets. We finished this phase by looking at interactions between the various aspects and actors of our system. The next phase of our report is the realization phase where we will look at aspects concerning the implementation of our system.

|  |
| --- |
| BOOK V |
| REALISZATION 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. Content

Content overview

INTRODUCTION

1. ARCHITECTURE OF THE APPLICATION
2. Logical architecture
3. Physical architecture
4. TECHNOLOGICAL CHOICE
5. Choice of programming language and framework
6. Choice of tools used

CONCLUSION

## INTRODUCTION

Here in the realization phase, we will concentrate on building or implementing our solution, based on the different analysis and conception that we had carried out, which will help to facilitate our work, this phase is as critical as the previous phases. We are going to look at the relationship that exist between the entities of the entity relational diagram. Furthermore, we will see the choices of technologies used for the implementation of our system.

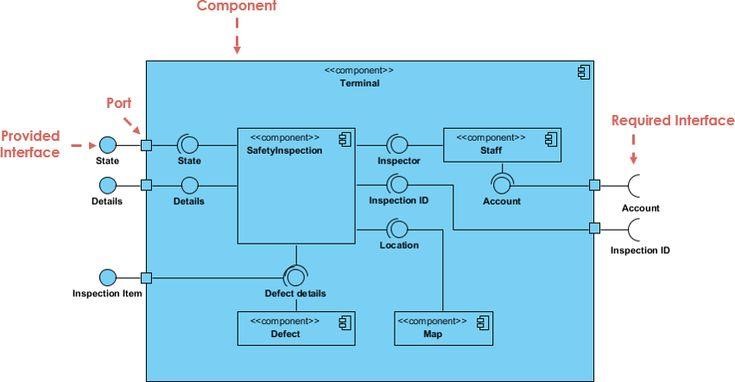
IMPLEMENTATION BRANCH

Here we will see the preliminary conception, detailed conception and documentation of the system.

1. PRELEMINARY DESIGN:
2. **Component Diagram:**
3. 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.

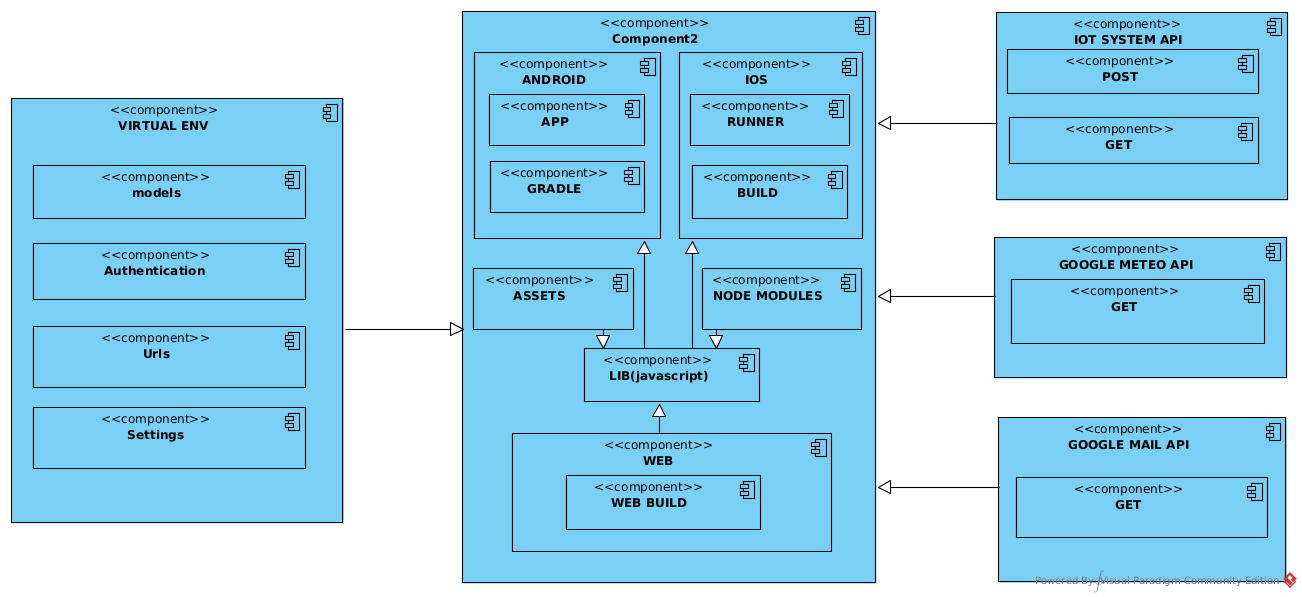
1. Formalism:



*Figure 20: Formalism of a component diagram (Source: https://www.pinterest.com/pin/551128073157994549/)*

*Table 17:Elements of a component diagram*

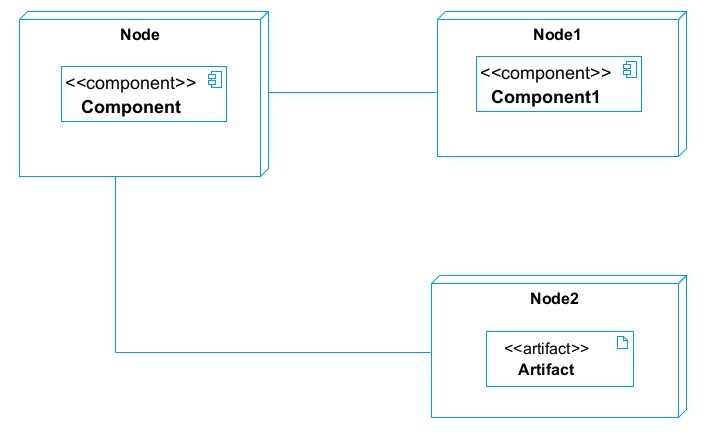
|  |  |  |
| --- | --- | --- |
| **NAME** | **REPRESENTATION** | **DESCRIPTION** |
| **A component** |  | A component is an abstract logical unit block of a system.it is represented as a rectangle with smaller rectangle in the upper right corner which saves as it icon for recognition. |
| **Dependency** |  | Dependency is a directed relationship which is used to show that some components are dependent on others for their correct functioning. |
| **interface** |  | An interface is a circle or a semicircle attached to a stick which looks like a lollipop. It describes groups of operations provided or required by components. |
| **port** |  | A port (represented by a small square at the end of a required or provided interface) is used when the components delegate the interfaces to an internal class. |



1. **Deployment diagram:**
   * 1. Definition:

consists of nodes. Nodes are nothing but physical hardware used to deploy the application 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

* + 1. Formalism



*Figure 25:Deployment Diagram Formalism*

*Figure 26:Deployment Diagram components*

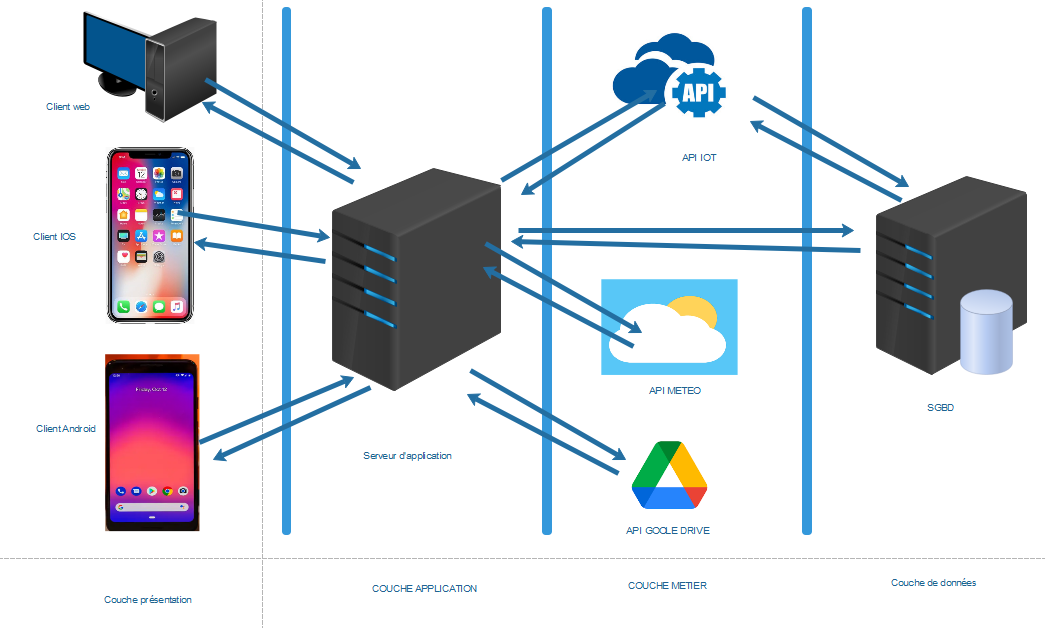
* 1. System Deployment Diagram

1. DETAILED DESIGN

## TECHNICAL BRANCH

1. GENERIC DESIGN
2. 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 it right proportion.



1. 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.

## CAPTURE TECHNICAL NEEDS

1. Physical architecture

The design of the DBMS depends on its architecture. An n-tier architecture partitions on the whole system into related but separated n modules, which can be independently modified, altered, changed or replace. A large amount of data on web servers, personal computers (pc) and others are link with networks with the help of basic client or server architecture.

Within the scope of our project, we made use of the multi-tier architecture This architecture separate it tiers from each other based upon the user and the manipulated data in the database. It is important to note that with the multi-tier architecture, only neighboring layers can communicate. Each layer has a well-defined communication interface and the evolution of the layer is independent of the other. The multi-tier of our system is made up of:

* The hardware tier, which represents
* The client tier, which is also known as our presentation inter-phase.
* Application Tier, which represents our web-server.

❖ The data tier, which represents our DBMS server

|  |  |  |
| --- | --- | --- |
| **Matériel** | **Fonction** | **Caractéristique** |
|  |  |  |
|  |  |  |

Softwares

|  |  |  |  |
| --- | --- | --- | --- |
| Logiciels | **Version** | **Rôle** | **Logo** |
| OS  WINDOWS  10 | 18 362 | Système d’exploitation nécessaire pour exécuter les logiciels. Ce système est celui sur lequel nous avons travaillé. |  |
| Sybase  PowerAMC | 16,5 | Atelier de génie logiciel utilisé pour la modélisation de la solution. Nous nous sommes servis de cet outil pour la modélisation des différents diagrammes de notre système. |  |
| Visuel studio code | 1.70.0.8 | Éditeur de texte utilisé pour saisir les lignes de code qui seront interprétées par le navigateur. |  |
| Arduino IDE | 1.8.19 | Éditeur de texte utilisé pour la programmation des cartes arduino |  |
| Postman | 9.4 | est une plateforme qui permet de simplifier chaque étape du cycle de vie des API et de rationaliser la collaboration, afin de créer, plus facilement et plus rapidement, de meilleures API |  |
| Draw.io | 20.2.8 | est une application en ligne accessible via le navigateur pour permettre la réalisation des diagrammes et le travail en groupe |  |
| Expo go | 2.25.2 | est un Framework et une plateforme pour les applications React, pour compiler les applications mobiles react directement dans le téléphone |  |
| Fritzing | 0.9.3 | est un logiciel libre de conception de circuits imprimés qui permet de concevoir de façon entièrement graphique le circuit et d’en imprimer le typon |  |
| Github | 3.0.6 | est un site web et un service de cloud qui aide les développeurs à stocker et à gérer leur code |  |
| Framework  react native | 0.69.0 | framework construit sur JavaScript pour créer des applications Android et iOS |  |
| Nest. js | 9.1.1 | **Est un framework de Node.** **js côté serveur**, idéal pour créer des applications backend hautement testables et maintenables |  |
| HEROKU | 7.63.4 | est un Paas (Platform as a service destiné au développement dans le cloud, permet le déploiement des d’applications web |  |
| EDRAW  MAX | 11.5.0 | est un logiciel de création de diagrammes techniques commerciaux 2D qui aide à créer des organigrammes, des organigrammes, des cartes mentales, des diagrammes de réseau, des plans d’étage, des diagrammes de flux de travail, des graphiques commerciaux et des diagrammes d’ingénierie |  |
| **PostgreSQL** |  | Système de gestion des bases de données |  |

### Technologies de développement

|  |  |  |
| --- | --- | --- |
| **Logo** | **Nom** | **Utilité** |
|  | Django – Framework | Utilisé pour le développement du noyau applicatif |
|  | JavaScript | Langage de programmation utilisé pour les traitements de  front-end |

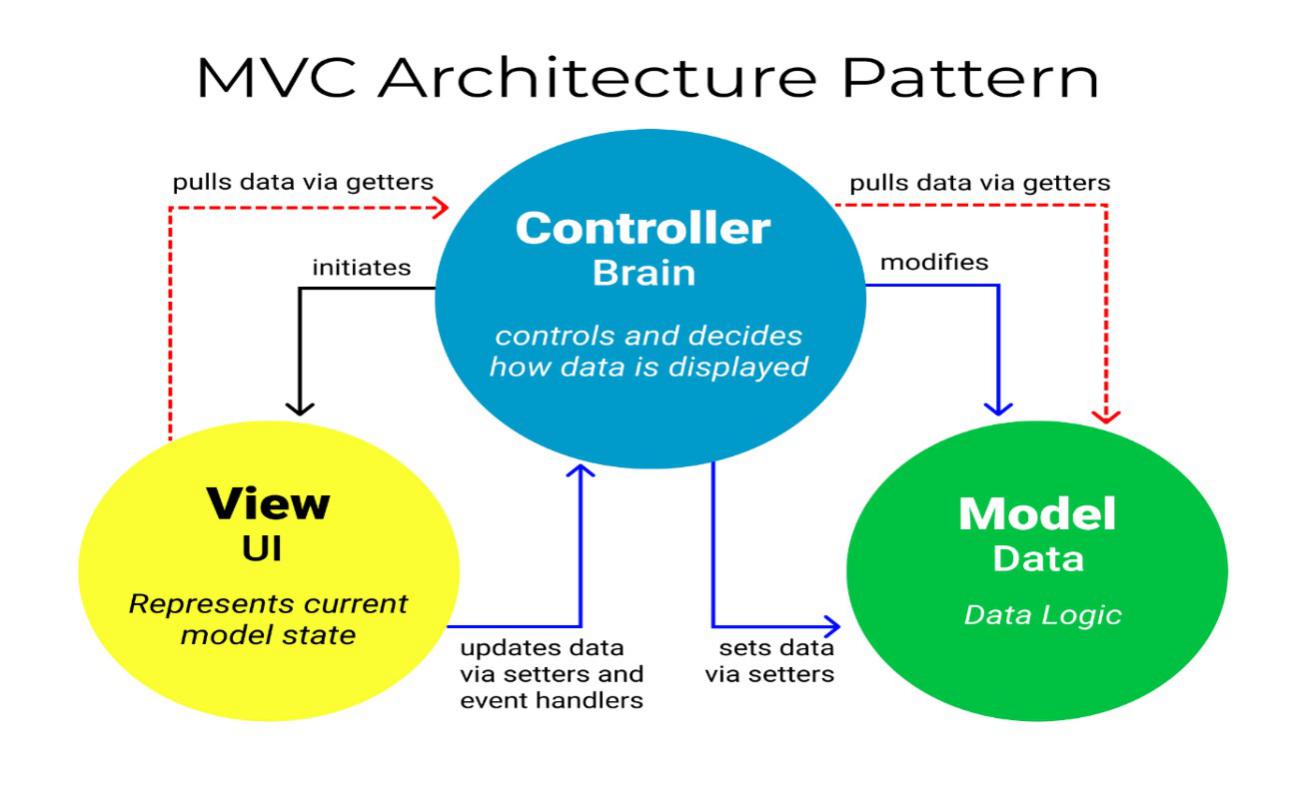
### Présentation de l’architecture logique du système IOT

1. Logical architecture:

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

* Model: The lowest level of the patten 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.

10 days

*Figure 19: The MVC architecture (Source: https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvcarchitecture-and-frameworks-explained/)*

|  |
| --- |
| BOOK VI |
| FUNCTIONAL TESTING |

|  |
| --- |
| BOOK VI |
| INSTALLATION 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 overview

INTRODUCTION

1. Class Diagram
2. State Transition Diagram
3. Package Diagram

CONCLUSION

BIBLIOGRAPHY

* Mrs. TCHINGA Alice UML course AICS-Cameroon 2022-2023
* Mr. MESSIO Ulrich PostgreSQL course AICS-Cameroon 2022-2023
* Mr. Ndifor Analogue Electronics course AICS-Cameroon 2022-2023
* OMG, OMG Unified Modelling Language 2.5, Object Management Group, March 2015, 794 pages.
* React-native course (from basics to advanced) from W3Uschool 2022.
* Arduino-IDE from arduinocc website 2023.
* JavaScript course from W3Uschool website 2023.

WEBOGRAPHY

• https://create.arduino.cc/projecthub/najad/interfacing-lcd1602-with-arduino-764ec4: which allowed us to connect an LCD screen [visited on June 13, 2022, at 1:31 PM]

• https://projetsdiy.fr/esp32-gpio-broches-fonctions-io-pwm-rtc-i2c-spi-adc-dac/amp/: which helped us understand the functioning of the ESP32 and its pins [visited on June 2, 2022, at 9:35 AM]

•https://m.youtube.com/watch?v=FdByUC12T-E&list=PLOYsAys6a6mkyPMmr8QVXtxgzLqd7GruN&index=1&t=0s: which allowed us to connect the LEDs to the ESP32 [visited on June 18, 2022, at 11:10 AM]

• https://www.raspberryme.com/esp32-http-get-et-http-post-avec-arduino-ide/: which enabled us to send POST requests from ESP32 to the API [visited on July 16, 2022, at 10:08 AM]

• https://arduino-self.over-blog.com/2020/12/un-controleur-de-servo.html: which helped us learn how to use the PCA [visited on July 6, 2022, at 2:08 PM]

• https://www.aranacorp.com/fr/utilisation-dun-module-pca9685-avec-raspberry-pi/amp/: which allowed us to understand how to connect the PCA and its role [visited on July 7, 2022, at 8:45 AM]

• https://arduinojson.org/v6/how-to/use-arduinojson-with-httpclient/: which enabled us to send requests from the IoT system to the API [visited on July 15, 2022, at 2:05 PM]

• https://www.youtube.com/watch?v=1Ssh2aTDPGY: which helped us understand interrupts [visited on July 18, 2022, at 4:24 PM]

• https://pythonprogramminglanguage.com/python-flask-websocket/: allowed us to study web-sockets

• https://randomnerdtutorials.com/esp32-esp8266-i2c-lcd-arduino-ide/: allowed us to study LCD in more detail [visited on July 29, 2022, at 5:03 PM]

• http://dspace.univ-tlemcen.dz/bitstream/112/6325/3/chapitre2.pdf: helped us better understand parking systems [visited on July 30, 2022, at 1:08 PM]

• https://www.slidescarnival.com/tag/tech/page/2?amp: helped us better organize our presentation [visited on August 5, 2022, at 11:11 AM]

# TABLE OF CONTENT

[DEDICATION 2](#_Toc1651824571)

[ACKNOWLEDGEMENT 2](#_Toc664903207)

[SUMMARY 4](#_Toc864887026)

[LIST OF FIGURES 5](#_Toc187245213)

[LIST OF TABLES 6](#_Toc302293075)

[GLOSSARY 7](#_Toc98850395)

[ABSTRACT 8](#_Toc2056011754)

[RESUME 9](#_Toc2091039671)

[GENERAL INTRODUCTION 10](#_Toc845700170)

[PART ONE: INSERTION PHASE 12](#_Toc1981378544)

[INTRODUCTION 14](#_Toc520486094)

[I. WELCOME AND INTEGRATION 15](#_Toc874079690)

[A. Welcome: 15](#_Toc1490508606)

[B. Integration: 15](#_Toc1405537492)

[II. GENERAL PRESENTATION OF ETECH 15](#_Toc134779082)

[G. Presentation of Etech: 16](#_Toc799779193)

[a. Geographical Location: 16](#_Toc948905812)

[b. Identification sheet: 16](#_Toc1993548896)

*[Table 1 : Identification sheet of Etech For Self-Worth](#_Toc39331239)* [16](#_Toc39331239)

[c. Hardware And Software Resources Of Etech 17](#_Toc1265557725)

[i. Hardware resources: 17](#_Toc818596165)

[The company has many pieces of equipment listed in the following table. 17](#_Toc1960796785)

*[Table 2 : Hardware resources of Etech For Self-Worth](#_Toc163911739)* [17](#_Toc163911739)

[ii. Software resources: 18](#_Toc1859928483)

*[Table 3: Software resources of Etech for Self-Worth](#_Toc950153860)* [18](#_Toc950153860)

[I. Visions and objectives: 19](#_Toc1296075204)

[A) Visions: 19](#_Toc1934445367)

[B) Objectives: 19](#_Toc1021402873)

[III. Activities, Organizational Chart, and Responsibilities of Etech For Self-Worth: 19](#_Toc136313773)

[A. Activities: 19](#_Toc731838746)

[B. Functional organization of Etech: 20](#_Toc553980704)

[C. Attributions: 20](#_Toc1788138344)

[1. General Management: 20](#_Toc1396741954)

[2. Web Programming Division: 21](#_Toc1418867730)

[3. Mobile Programming Division: 21](#_Toc1975383557)

[4. IOT and Artificial intelligence Division: 21](#_Toc1699035029)

[5. Secretary: 21](#_Toc1517718125)

[For billing, photocopies, customer registration, and printing. 21](#_Toc1883911663)

[CONCLUSION 22](#_Toc1642591052)

[PART TWO: TECHNICAL PHASE 23](#_Toc215934648)

[IV. 23](#_Toc1717806560)

[BOOK I 24](#_Toc15593498)

[I. EXISTING SITUATION 24](#_Toc1090014338)

[INTRODUCTION 26](#_Toc1060831518)

[DESCRIPTION OF THE EXISTING SYSTEM 27](#_Toc1421130990)

[CONCLUSION 29](#_Toc1224793420)

[BOOK II 30](#_Toc1860610711)

[II. CONTEXT AND JUSTIFICATION 30](#_Toc222553154)

[INTRODUCTION 32](#_Toc1070858668)

[I. CONTEXT AND JUSTIFICATION OF STUDIES 33](#_Toc1899941950)

[II. PROBLEM DEFINITION 33](#_Toc1488110879)

[III. OBJECTIVES 35](#_Toc1889454834)

[IV. EXPRESSION OF NEEDS 36](#_Toc1713255087)

[V. PLANNING THE PROJECT 39](#_Toc1652022618)

[VI. ESTIMATE OF THE PROJECT 42](#_Toc1601899669)

[VII. CONSTRAINTS 43](#_Toc515925299)

[VIII. DELIVRABLES 44](#_Toc800614174)

[CONCLUSION 45](#_Toc1388861389)

[BOOK III 46](#_Toc1537328172)

[IX. ANALYSIS PHASE 46](#_Toc936927947)

[INTRODUCTION 48](#_Toc2120700135)

[I. PRESENTATION OF ANALYSIS METHOD 49](#_Toc2091308877)

[A. MODELING WITH UML (2.5) 50](#_Toc577582643)

[a. Structural Diagrams: 50](#_Toc1369958441)

[b. Behavioral Diagrams: 50](#_Toc1362692959)

[c. UML 2.5 diagrams overview : 51](#_Toc405482552)

[B. The Process of Development of a Software 51](#_Toc921509822)

[a. A Unified Process : 51](#_Toc732927437)

[b. The Two Track Unified Process (2TUP) 52](#_Toc141910568)

[a) The left branch (functional branch) 53](#_Toc416617226)

[b) The right branch (Technical branch) 53](#_Toc948862085)

[c) The middle branch (Realization or implementation branch) 53](#_Toc1859717128)

[JUSTIFICATION OF METHOD OF ANALYSIS 55](#_Toc432210724)

[II. MODELING OF THE SYSTEM 56](#_Toc2038876423)

[Capture Of Functional Needs: 56](#_Toc773064998)

[1. Use Case Diagram: 56](#_Toc1853341714)

[a) Definition: 56](#_Toc1116186196)

[b) Formalism: 56](#_Toc486192061)

[c) The Actors Of Our System 59](#_Toc2075894868)

[d) General Use Case Diagram 60](#_Toc39561216)

[3. Sequence Diagram: 64](#_Toc238650363)

[4. Activity Diagram 66](#_Toc1416522099)

[a) Definition: 66](#_Toc1929016050)

[b) Formalism 67](#_Toc1951905450)

[BOOK IV 74](#_Toc921061069)

[III. CONCEPTION PHASE 74](#_Toc1383432072)

[INTRODUCTION 76](#_Toc320347102)

[a) Definition 80](#_Toc1721675243)

[b) Formalism: 80](#_Toc624809813)

[2. Package Diagram: 82](#_Toc1857675274)

[CONCLUSION 84](#_Toc511119542)

[BOOK V 85](#_Toc598026300)

[IV. REALISZATION PHASE 85](#_Toc1801500503)

[INTRODUCTION 86](#_Toc1088702185)

[2. Deployment diagram: 88](#_Toc1967984742)

[A. DETAILED DESIGN 88](#_Toc1016709815)

[I. TECHNICAL BRANCH 89](#_Toc1494184738)

[A. GENERIC DESIGN 89](#_Toc742010916)

[i. Hardware diagram of the system: 89](#_Toc1749637252)

[CAPTURE TECHNICAL NEEDS 89](#_Toc1636095306)

[i. Physical architecture 89](#_Toc1158628143)

[Technologies de développement 92](#_Toc551015689)

[Présentation de l’architecture logique du système IOT 92](#_Toc1348328786)

[ii. Logical architecture: 92](#_Toc1590838867)

[II. IMPLEMENTATION BRANCH 94](#_Toc442408464)

[C. PRELEMINARY DESIGN: 94](#_Toc2121393784)

[1. Component Diagram: 94](#_Toc1296696934)

[BOOK VI 97](#_Toc1558594660)

[III. FUNCTIONAL TESTING 97](#_Toc460102197)

[BOOK VI 98](#_Toc1225108154)

[IV. INSTALLATION AND USER GUIDE 98](#_Toc1598155877)

[BIBLIOGRAPHY 100](#_Toc698752560)

[WEBOGRAPHY 101](#_Toc494146606)

[TABLE OF CONTENT 102](#_Toc1379688279)

|  |
| --- |
|  |
|  |