

REPUBLIC OF CAMEROON

Peace - Work - Fatherland

REPUBLIQUE DU CAMEROUN

Paix – Travail - Patrie



**AFRICAN INSTITUTE OF
COMPUTER SCIENCES -
CAMEROON**

PAUL BIYA TECHNOLOGICAL

P.O. Box: 13719 Yaoundé

[Tel: +237.242.729.957](tel:+237.242.729.957) Fax: 22729958

E-mail: contact@iaicameroun.com

**INSTITUTE AFRICAINE DE
L'INFORMATIQUE-CAMEROUN
CENTRE TECHNOLOGIQUE DE
L'EXCELLENCE PAUL BIYA**

P.O. Box: 13719 Yaoundé

[Tel: +237.242.729.957](tel:+237.242.729.957) Fax: 22729958

Courriel : contact@iaicameroun.com

PERSONALIZED PROJECT REPORT

**THEME:
DEVELOPMENT OF AN EXPENSE TRACKER APP**

Written by:

WEMAYI TEMOUAFO KEVIN EVRARD

Level III Software Engineering student

Supervised by:

Mr Agbor Anderson

Computer Engineering Lecturer at

AICS-Cameroon

Academic Year 2023 - 2024



DEVELOPMENT OF AN EXPENSE TRACKER APP





DEVELOPMENT OF AN EXPENSE TRACKER APP



DEDICATION

THIS WORK IS DEDICATED TO MY FAMILY. FOR THIER LOVE, SUPPORT AND
ENCOURAGEMENT TOWARDS MY ACADEMIC SUCCESS.



ACKNOWLEDGMENTS

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.
- Our academic supervisor **Mr. AGBOR Anderson** for her moral support, effort and advices throughout my personal project.
- To all my teachers of **AICS CAMEROON** for the knowledge they affected me with for me to be where I am today.
- The countless contributors of open source programming community, for their great help in learning basic skills and detecting and solving bugs.
- To all my classmates for their collaborative work throughout the academic year.



SUMMARY

DEDICATION	ii
ACKNOWLEDGMENTS	iii
SUMMARY	iv
LIST OF ABBREVIATIONS	v
LIST OF FIGURES	vi
LIST OF TABLES	viii
ABSTRACT	ix
RESUME	x
GENERAL INTRODUCTION.....	1
PART ONE: EXISTING SYSTEM.....	2
PART TWO: SPECIFICATION BOOK	11
PART FOUR: ANALYSIS PHASE	23
PART FIVE: CONCEPTION PHASE	54
PART SIX: REALIZATION PHASE	68
PART SEVEN: USER GUIDE.....	78
PERSPECTIVES	87
GENERAL CONCLUSION	88
BIBLIOGRAPHY	XII
WEBOGRAPHY	XIII
TABLE OF CONTENT	XIV



LIST OF ABBREVIATIONS

- 2TUP: Two Track Unified Process
- AICS: African Institute of Computer Sciences
- API: Application Programming Interface
- CSS: Cascading Style Sheets
- DBMS: Database Management System
- HTML: Hypertext Markup Language
- JS: JavaScript
- MVT: Model View Template
- SQL: Structured Query Language
- UML: Unified Modelling Language



LIST OF FIGURES

Figure 1 UML diagrams overviews (Source: https://creatly.com/log/diagram-types-examples/)	29
Figure 2 : 2TUP diagram (Source: https://www.mysciencework.com/omniscience/pervasive-mobile-healthcare-system-based-on-cloud-computing)	31
Figure 3 formalism of a use case diagram	34
Figure 4 General use case	38
Figure 5 use case view expense	39
Figure 6 Communication diagram formalism	41
Figure 7 communication diagram employee enrollment	42
Figure 8 communication diagram register	43
Figure 9 add category communication diagram	43
Figure 10 Formalism of a sequence diagram	44
Figure 11 Sequence diagram add category	46
Figure 12 sequence diagram register	47
Figure 13 Formalism of an activity diagram	48
Figure 14 Authentication activity diagram	51
Figure 15 activity diagram add category	52
Figure 16 Formalism of a class diagram	59
Figure 17 class diagram of spendwise	60
Figure 18 Class diagram formalism	61
Figure 19 state machine diagram for add category	63
Figure 20 Package diagram formalism	64
Figure 21 spend wise package diagram	67
Figure 22 deployment diagram of spendwise	72
Figure 23 Formalism of a component diagram	73



DEVELOPMENT OF AN EXPENSE TRACKER APP



Figure 24 spendWise component diagram.....	75
Figure 25 pycharm installation	81
Figure 26 : Launching the installation wizard	82
Figure 27 Clicking on next	82
Figure 28 finish	83
Figure 29 Register form	84
Figure 30 login form	84
Figure 31 dashboard.....	85
Figure 32 admin page.....	85



LIST OF TABLES

Table 3 Hardware resources.....	18
Table 4 Software resources	18
Table 5 Human Resources	20
Table 6 Total Estimated Cost.....	20
Table 8 Components of a use case diagram.....	35
Table 9 Identification of actors and their use cases	37
Table 10 Textual description of use case Authenticate	39
Table 12 Components of a communication diagram	42
Table 13 Component of a sequence diagram	45
Table 14 Components of an activity diagram	49
Table 15 components of a package diagram.....	65
Table 16 Component of a component diagram.....	74



ABSTRACT

Effective financial management is crucial in both personal and professional spheres. Our expense tracker app addresses this need by providing a comprehensive solution for users to monitor and control their expenditures effortlessly. With a focus on simplicity and functionality, the app allows individuals and businesses alike to track expenses, categorize transactions, and analyze spending patterns in real-time. Advanced features such as customizable budgeting tools and insightful financial reports empower users to make informed decisions, whether managing personal finances or overseeing business expenditures. Designed with user convenience and data security as top priorities, our app ensures a seamless experience for anyone seeking to improve their financial well-being and efficiency.

Key words:

1. Expense tracker app
2. Financial management
3. Real-time
4. Budgeting tools



RESUME

La gestion efficace des finances est cruciale tant dans la sphère personnelle que professionnelle. Notre application de suivi des dépenses répond à ce besoin en offrant une solution complète aux utilisateurs pour surveiller et contrôler leurs dépenses sans effort. Axée sur la simplicité et la fonctionnalité, l'application permet aux particuliers et aux entreprises de suivre les dépenses, de catégoriser les transactions et d'analyser les habitudes de dépenses en temps réel. Des fonctionnalités avancées telles que des outils de budget personnalisables et des rapports financiers détaillés permettent aux utilisateurs de prendre des décisions éclairées, que ce soit pour la gestion de leurs finances personnelles ou le suivi des dépenses professionnelles. Conçue avec la commodité des utilisateurs et la sécurité des données comme priorités absolues, notre application garantit une expérience fluide à quiconque souhaite améliorer son bien-être financier et son efficacité.

Mots cle :

1. application de suivi des dépenses
2. gestion financière
3. en temps réel
4. outils de budget



GENERAL INTRODUCTION

Managing personal and business expenses is a fundamental aspect of financial stability and efficiency. Whether for individuals seeking to budget effectively or businesses aiming to optimize financial resources, the ability to track expenditures accurately plays a pivotal role. In recent years, advancements in technology have revolutionized the way expenses are managed, shifting from traditional methods to sophisticated digital solutions. The rise of “SPENDWISE“ an app and software has streamlined this process, offering users intuitive tools to monitor spending, categorize transactions, and analyze financial trends in real-time. These tools not only enhance financial awareness but also empower users to make informed decisions, improve savings strategies, and ensure compliance with financial goals and regulations. As the digital landscape continues to evolve, the integration of expense tracking technology proves indispensable in fostering financial accountability and maximizing resource allocation across diverse sectors and personal finance domains.

- **Specification book:** The specification book is the part of the document that describes exactly what is expected from us taking into consideration the time and the cost of the project. In other to proposed a suitable solution for the user.
- **Analysis phase:** It presents the analysis method chosen with presentation of all diagrams use for the analysis of the project.
- **Conception phase:** This phase enables us to model the solution based on the criticism done at the analysis phase.
- **Realization phase:** This phase contains the implementation of analyzed and conceived system.
- **User guide:** Contains instruction how to use the deployed system.



PART ONE: EXISTING SYSTEM



Preamble

In this chapter, we shall give a detailed explanation of our theme. That is, what the theme is all about and to give more precision concerning the application we are to produce. This chapter will also describe the problematic that led to our proposed solution.

Overview

INTRODUCTION

- I THEME PRESENTATION
- II STUDY OF THE EXISTING SYSTEM
- II CRITICISM OF THE EXISTING SYSTEM
- IV PROBLEMATIC
- V PROPOSED SOLUTION
- CONCLUSION



INTRODUCTION

The existing system part of our report is very important. Here, we shall present our theme to improve our understanding of it and increase the likelihood of succeeding, present the study of the existing system, which will guide us on the approach to take. After studying the existing system, we shall present our different critics concerning it. We shall also give the problematic i.e., the different problems phased by the existing system and finally we shall propose a solution to solve these problems.



I. THEME PRESENTATION

A theme is a term use to define something. It help to delimitate a scope of work. In our case, we shall work on the theme “**DEVELOPMENT OF AN EXPENSE TRACKER APP**” wit. As this theme implies, we shall produce a platform that focuses mostly on the management of expenses.



II. STUDY OF THE EXISTING SYSTEM

The traditional system of expense tracking typically involves manual methods and basic tools that may include:

1. **Paper-Based Records:** Many individuals and businesses still rely on notebooks, spreadsheets, or paper receipts to record expenses manually. This method requires meticulous data entry and organization.
2. **Excel Spreadsheets:** Excel or similar spreadsheet software is commonly used to create tables where expenses are logged, categorized, and calculated manually. Users input transaction details such as date, amount, category, and description.
3. **Receipts and Invoices:** Physical receipts and invoices are collected and stored for later entry into the expense tracking system. This can lead to delays in recording and potential loss of documentation.
4. **Bank Statements:** Some rely on reviewing bank statements periodically to categorize expenses. This method may not provide real-time insights into spending patterns.
5. **Manual Calculation and Analysis:** Users manually calculate totals, review spending trends, and analyze budget adherence based on the recorded data. This process can be time-consuming and prone to human error.

Overall, while these methods have been effective to a certain extent, they are labor-intensive, lack real-time updates, and may not provide robust analysis capabilities needed for effective financial management in today's fast-paced digital world.



III. CRITICISM OF THE EXISTING SYSTEM

Because of the study of the existing system of the expenses management, we came out with the following weaknesses:

1. **Manual Data Entry Errors:** The reliance on manual entry increases the risk of errors such as typos, misclassifications, and duplicate entries, leading to inaccurate financial records.
2. **Lack of Real-Time Updates:** Traditional methods often do not provide real-time updates on expenses, resulting in delays in accessing current financial information and insights.
3. **Time-Consuming Processes:** Managing expenses manually is labor-intensive and time-consuming, requiring significant effort to input data, reconcile receipts, and calculate totals.
4. **Limited Accessibility:** Access to expense records and reports may be restricted to specific individuals or departments, hindering transparency and collaboration within organizations.
5. **Difficulty in Analysis:** Analyzing spending trends and generating meaningful reports from manual records can be complex and may not provide detailed insights needed for effective financial decision-making.
6. **Risk of Loss and Security Issues:** Paper receipts and documents can be easily lost, damaged, or misplaced, posing security risks and leading to gaps in expense tracking and financial data management.

These criticisms underline the inefficiencies and challenges associated with traditional expense tracking methods, emphasizing the need for modern, digital solutions that offer real-time updates, automation, and enhanced security.



IV. PROBLEMATIC

The study of the functioning of the existing system has permitted us to identify certain critics. As a result of those different critics, there major problem is **“How to ease the management of expenses”**?



V. PROPOSED SOLUTION

In order to solve the problems stated above, we are to realize an application based on the following points:

1. Automated Data Entry: Implementing automatic data entry features that capture and categorize expenses in real-time, reducing the risk of errors associated with manual input.
2. Real-Time Updates: Providing instant updates on expenses, balances, and budget status to ensure users have current financial insights whenever they need them.
3. Simplified Expense Management: Offering a user-friendly interface with intuitive tools for easy expense recording, categorization, and management, streamlining the entire process.
4. Secure Cloud Storage: Utilizing secure cloud storage to store expense data safely, reducing the risk of loss or theft associated with paper receipts and physical documents.
5. Customizable Reporting: Enabling users to generate customizable reports and analytics on spending trends, budget adherence, and financial performance, supporting informed decision-making.
6. Integration with Financial Tools: Integrating seamlessly with other financial management tools and platforms to provide a holistic view of finances and enhance overall efficiency in financial management.

These solutions aim to enhance accuracy, efficiency, and accessibility in expense tracking, empowering users to manage their finances more effectively and make informed financial decisions.



CONCLUSION

The theme is a vital component of the project as it helps to define the scope of work and identify the system under consideration. The study of the existing system is equally crucial. This is because, in order to improve a system, it is necessary to thoroughly examine it, identify its existing problems and limitations, and critically analyze its shortcomings. Once the study is completed and the problematic areas are identified, proposed solutions can be developed to address and resolve the issues that have been discovered.



PART TWO: SPECIFICATION BOOK



Preamble

The specification book outlines the goals to be achieved through this project and the responsibilities of the different parties involved in the project. It specifies and describes the subject and the needs of the users, as well the conditions necessary to realize the project.

Content Overview

INTRODUCTION

- I. CONTEXT AND JUSTIFICATION
- II. OBJECTIVES OF THE PROJECT
- III. EXPRESSION OF NEEDS
- IV. ESTIMATED COST OF THE PROJECT
- V. PROJECT PLANNING
- VI. LIST OF PARTICIPANTS

CONCLUSION



INTRODUCTION

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



I. CONTEXT AND JUSTIFICATION

Effective management of personal finances is a universal challenge, and Cameroonians are no exception. In Cameroon, as in many parts of the world, individuals and businesses face the ongoing task of budgeting, tracking expenses, and ensuring financial discipline. However, traditional methods of managing finances, such as manual record-keeping or basic spreadsheets, often fall short in providing real-time insights and comprehensive financial analysis. Cameroon, with its diverse economic landscape ranging from bustling urban centers to rural communities, presents unique challenges in financial management. For urban professionals navigating daily expenses in cities like Douala and Yaoundé, as well as entrepreneurs and small business owners in regions like Bamenda and Buea, the need for a practical, accessible tool to track expenditures is crucial. An “EXPENSE TRACKER APP” tailored for Cameroonians can address these challenges by offering innovative features.



II. OBJECTIVES OF THE PROJECT

A. General objective of the project

The general objective of the project is to develop an expense tracker app tailored for Cameroonians. It aims to provide a user-friendly platform for real-time expense tracking, localized currency support, customizable budgeting tools, and enhanced financial literacy, all while ensuring security and privacy.

B. Specific objectives

Specifically, it will consist of:

- The user will have the possibility to add his/her expenses
- The user will have the possibility to add his income
- The user will be able to print his expenses reports

;



III. EXPRESSION OF NEEDS

A. Functional needs

Functional needs describe what the users can do on the system. There are, listed below as follows:

- Add expense;
- Add income;
- Add source;
- Add category;
- Remove category;

The above functionalities are explained below:

- **Add expense:** The user will be able to add expenses
- **Add category:** The user will be able to add any category in case he have new things he spends his money on.
- **Add Income:** The user can add his income being it daily, month or weekly.
- **Add source:** The user will be able to track the sources of each income he/she gets.



B. Non-functional Needs

I. Hardware requirements

a. Portability and ease of installation:

The hardware should be lightweight and easily transportable; it should require little or no skill to install and setup guide.

b. Reusability and Reliability:

The servers should be re-usable

II. Software requirements

a. Security:

Since our system manages employee information, a good mechanism should be implemented to give access to the system just to authorized users.

b. Performance:

The software should provide a user-friendly interface. It should easily and readily connect to the hardware.



IV. ESTIMATED COST OF THE PROJECT

A. Hardware Resources

Table 1 Hardware resources

RESOURCES	HARDWARE	USAGE	QUANTIT Y	UNIT COST(FCFA)
COMPUTER	DELL Windows 10 pro 64 bits, Latitude E6520, 8GB RAM, 500GB hard disk; intel core i5.	main resource for the accomplishment of this project; report writing, designs, analysis, coding	1	300, 000
Removable disk	8GB USB key	For file transfer from one computer to another.	1	8000
CD	CD-ROM		3	2500
Local Server				
MODEM	4g LTE-advanced <i>mobile</i> <i>WiFi</i> hotspot <i>mq531</i> 150mp s	Used to share internet access amongst devices	1	15000
Total				325500

B. Software Resources

Table 2 Software resources



DEVELOPMENT OF AN EXPENSE TRACKER APP



RESOURCES	HARDWARE	USAGE	QUANTITY	UNIT COST(FCFA)
Development tool	Visual studio code	Code editing	1	Open source
Project planner	Gantt project	Project planning tool	1	Free software
Operating system	Microsoft windows 10 pro	Computer operating system	1	103000
Text Editor	Microsoft office word 2019	For typing and formatting of report	1	161000
Presentation	Microsoft office PowerPoint 2019	For making presentations	1	84230
Modelling tool	Visual-paradigm Modeler version	Modelling the system in uml	1	3350
Geo-location plan designing tool	Microsoft Visio 2007	To draw location plan	1	Free version
Web browser	Google chrome	For running and testing application	1	Free version
Database management system	PostgreSQL	Communing with the database	1	Open source



DEVELOPMENT OF AN EXPENSE TRACKER APP



Total				351,580
--------------	--	--	--	---------

C. Human Resources

Table 3 Human Resources

RESOURCES	NUMBER	COST PER DAY	NUMBER OF DAYS	COST (FCFA)
Project manager	1	150000	3	450000
Analyst	1	150000	2	300000
Designer	1	100000	5	500000
Programmer	1	100000	5	500000
Security expert	1	100000	2	200000
Program Tester	1	20000	1	20000
Total	7	890000		2470000

D. Total Project Estimated Cost

Table 4 Total Estimated Cost

Overall total (FCFA)
2470000



LIST OF PARTICIPANTS DELIVERABLES

A. List of participants

NAME	TITLE	ROLE
Mr. Agbor Anderson	Lecturer at AICS-CAMEROON	Academic supervisor
WEMAYI TEMOUAFO KEVIN EVRARD	software engineering student at AICS-Cameroon	Analyst and developer

B. Deliverables

We are supposed to submit a report at the end of this project that includes;

- A specification book;
- An analysis book;
- A conception book;
- A realization book;
- A user guides.



CONCLUSION

The specification book helps us to site the different needs we need in order to implement our mobile application for student follow-up, 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.

Content Overview

INTRODUCTION

- I. PRESENTATION OF THE ANALYSIS METHOD
 - II. JUSTIFICATION OF THE ANALYSIS METHOD
 - III. MODELING OF THE SYSTEM
- ### 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.



II. PRESENTATION OF THE ANALYSIS METHOD

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;

Merise: It is an information system design and development widely used in France. It was first early introduced in 1980s.

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.

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.

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.



A. MODELING WITH UML 2.5

The unified modeling language (UML) is a general purpose, developmental modeling language in the field of computer engineering that is intended to provide a standard way to visualize the design of a system. It was born from the merge of the three object-oriented programming methods namely:

The **BOOCH Method** (a method for object-oriented software development that has an object 3 modeling language, an iterative object-oriented development process and a set of recommended practices) developed by Grady brooch, James Rumbaing's **Object Modeling Technique** (OMT) (an object modeling approach for software modeling and design) and Ivar Jacobson's **OBJECT SOFTWARE ENGINEERING** (OOSE) (software design technique used in object-oriented programming). UML was created to forge a common, semantically and syntactically rich visual modeling language for the architecture, design and implementation of complex software systems both structurally and behaviorally.

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.

a. 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.

b. 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:



DEVELOPMENT OF AN EXPENSE TRACKER APP



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

B. UML 2.5 diagrams overview

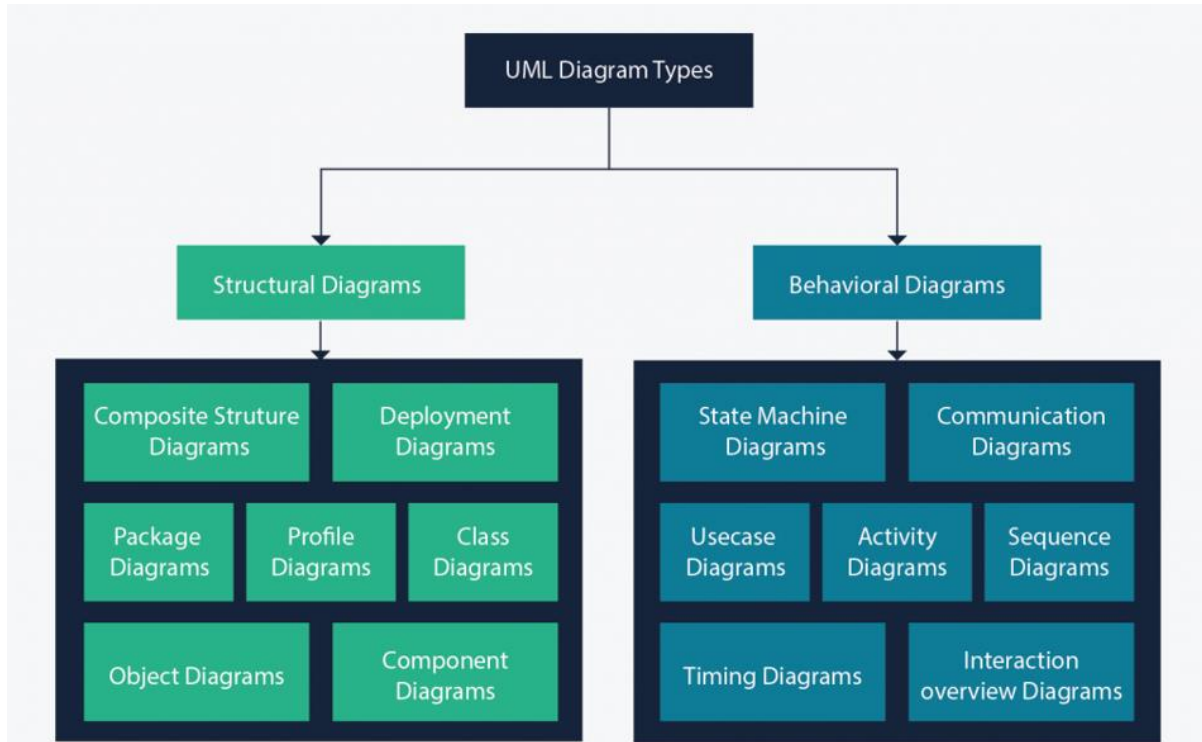


Figure 1 UML diagrams overviews (Source: <https://creately.com/log/diagram-types-examples/>)

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.



C. 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.

D. A Unified Process

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

Iteration are 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 users evaluate the evolution of the system.

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.

E. 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.

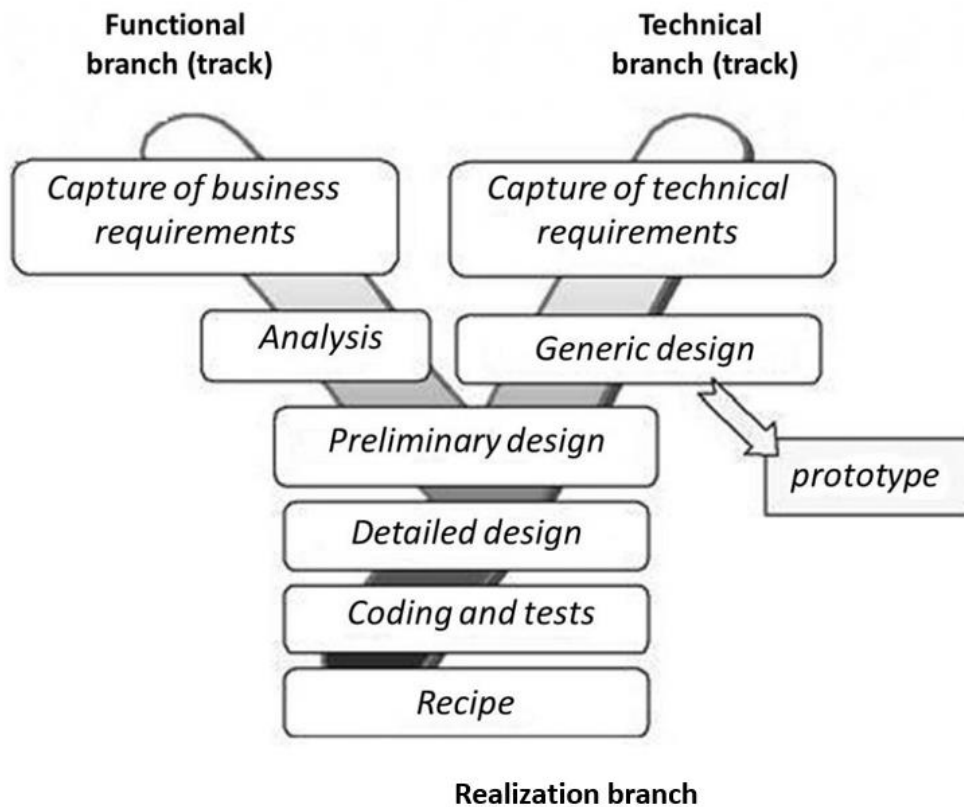


Figure 2 : 2TUP diagram (Source: <https://www.mysciencework.com/omniscience/pervasive-mobile-healthcare-system-based-on-cloud-computing>)



a. The left branch (functional branch)

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

b. The 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 is the phase where we program the designed features and test the coded features.

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



III. JUSTIFICATION OF THE ANALYSIS METHOD

Some developers have been successful at developing small systems without the use of diagrams or other features of modelling languages. However, as their systems become larger and larger, such developers have increasingly difficulties to seeing the 'big picture' of their system and are liable to create poor designs and take much longer time in their work. Most system are therefore documented with the use of diagrams. These provide views of structure and functionality that would be difficult to grasp by looking at code or textual description alone. In other words, diagrams provide abstraction.

Our choice for the UML modelling language and the 2TUP development process are based on the following criteria:

- UML is a language which is centered on user's needs
- UML is based on the object-oriented approach
- 2TUP offers a development cycle (the Y shape development cycle) which dissociates the technical aspects from the functional aspects
- UML produces good standards for software development
- UML has large visual elements to construct and easy to follow
- 2TUP is a process based on object approach and is constructed on UML.

IV. MODELLING OF THE PROPOSED SOLUTION

A. Capture of the functional needs

The functional requirements of a system specify the desired behavior or basic functionalities that the system is required to perform. In this section, we will present the following diagrams: a use case diagram along with textual descriptions of use cases, an activity diagram, a communication diagram, an interaction overview diagram, and a state machine diagram.

1. Use Case Diagram

a. Definition

A use case diagram is the primary form of a system/ software requirements for a new software program under development. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). Use cases once specified can be donated both textual and visual representation (such as UML). A key concept of use case modeling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior.

b. Formalism

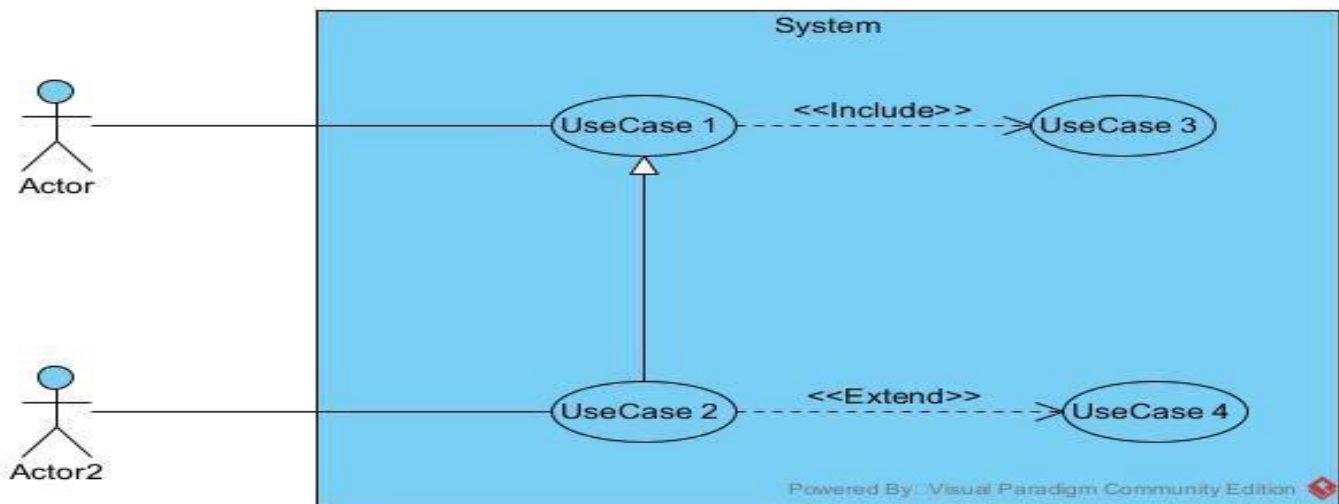

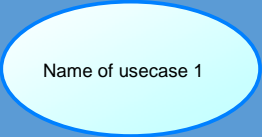
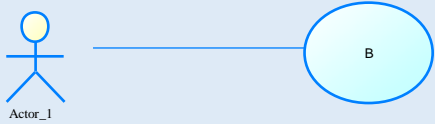
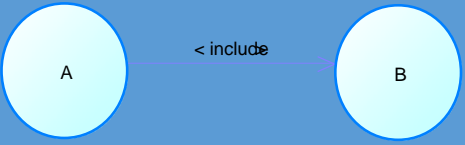


Figure 3 formalism of a use case diagram



c. Components of a use case diagram

Table 5 Components of a use case diagram

Element	Description and Main properties	Notation
Actor	Represents an entity that directly interacts with the system. The actor is what performs the different possible actions of the system.	
Use case	It corresponds to the objective/function of the system motivated by the need for one or more actors. The set of use cases describes the goal of a system. They are described in the form of infinitive verbs plus objects.	
Association	It indicates that an actor takes part in a use case.	
Inclusion	An inclusion relation denotes that an included action must be performed before another action can be performed. A includes B signifies that B is a compulsory part of A.	



DEVELOPMENT OF AN EXPENSE TRACKER APP



Exclusion	It denotes that an action may be performed while another one is being performed. A use case B extends A means B is an optional part of A.	
Inheritance	It is the only possible relation between actors.	
Generalization	This shows that a uses case is a kind of another. This relation also permits to decompose a complex case into smaller and simple cases.	
System	This is the representation of our system	



d. Identification of actors and their use cases

We will now list the actors likely to interact with the system, but at first, we will define an actor.

An actor represents the abstraction of a role played by external entities that interact directly with the system studied. The actors of the system identified at first are:

Table 6 Identification of actors and their use cases

Actor	Role
Employee	Their responsibility is to check in/out and view their presence history.
Manager	He is in charge of recording attendance of the employee, managing their presence on real time and manage employee leave request adding and removing employees in the system
Administrator	The administrator is in charge of the general functioning of the system.
Fingerprint Sensor	Its role is to register each employee uniquely and to identify a random employee.

e. General use case diagram

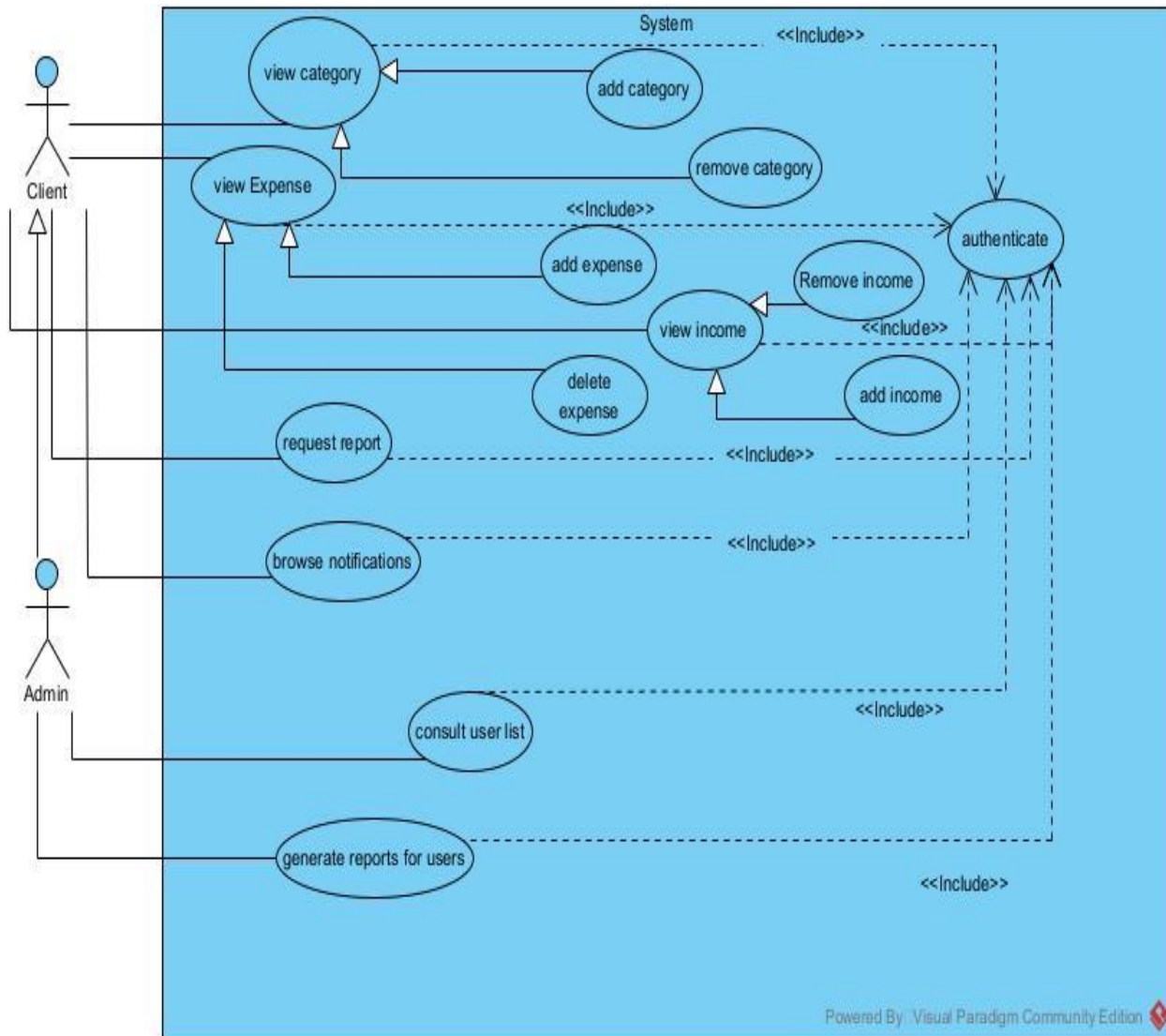


Figure 4 General use case

f. Use case diagram View expense

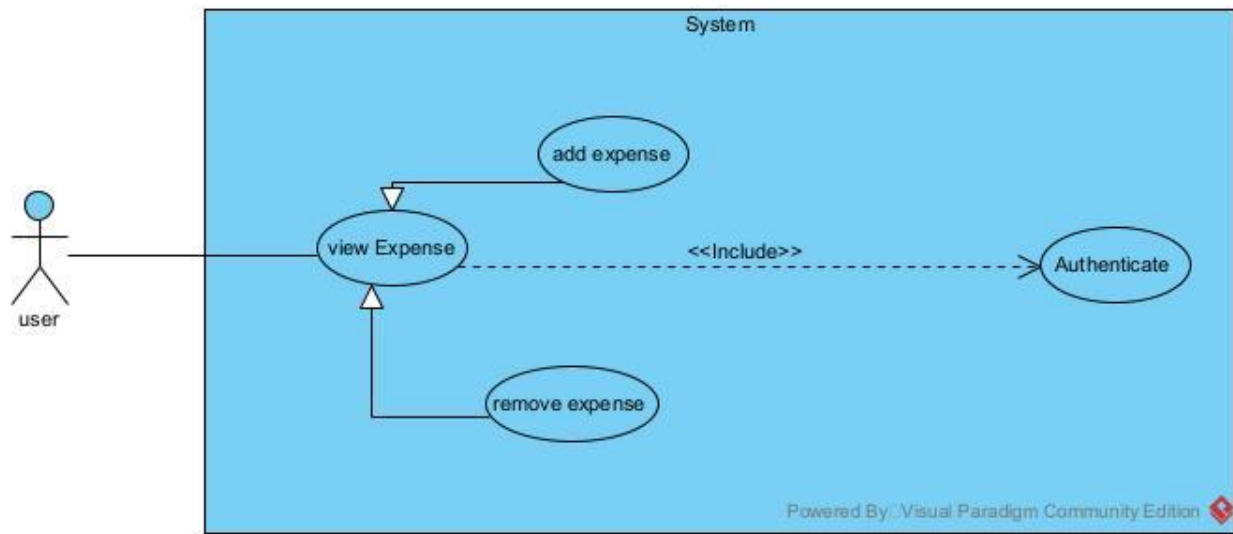


Figure 5 use case view expense

❖ Use case textual description

As we mentioned earlier, a use case represents a transaction. Here, it is a question of breaking down the use cases, making a textual description and then emphasizing on the different detailed scenario. This will be done using diagrams known as the sequence diagram and the activity diagram.

a. Use case authenticate

Table 7 Textual description of use case Authenticate

Goal in context:	Grant the user access to the application
Pre-condition(s):	The system should be loaded
Nominal scenario:	<ol style="list-style-type: none"> 1. The user click the login button 2. The system provides the login form 3. The user fill and submit the form 4. The system verifies the credentials 5. The system verifies the information in the database 6. The system start the user's session



DEVELOPMENT OF AN EXPENSE TRACKER APP



Alternative scenario (invalid credentials)	4.1 The system display the error message 4.2 The system returns to step (2) of the nominal scenario
Alternative scenario (user does not exist)	5.1 The system display the error message 5.2 The system returns to step (2) of the nominal scenario
Post-conditions (s)	The user's session is launched.

1. Communication Diagram

a. 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.

b. Formalism

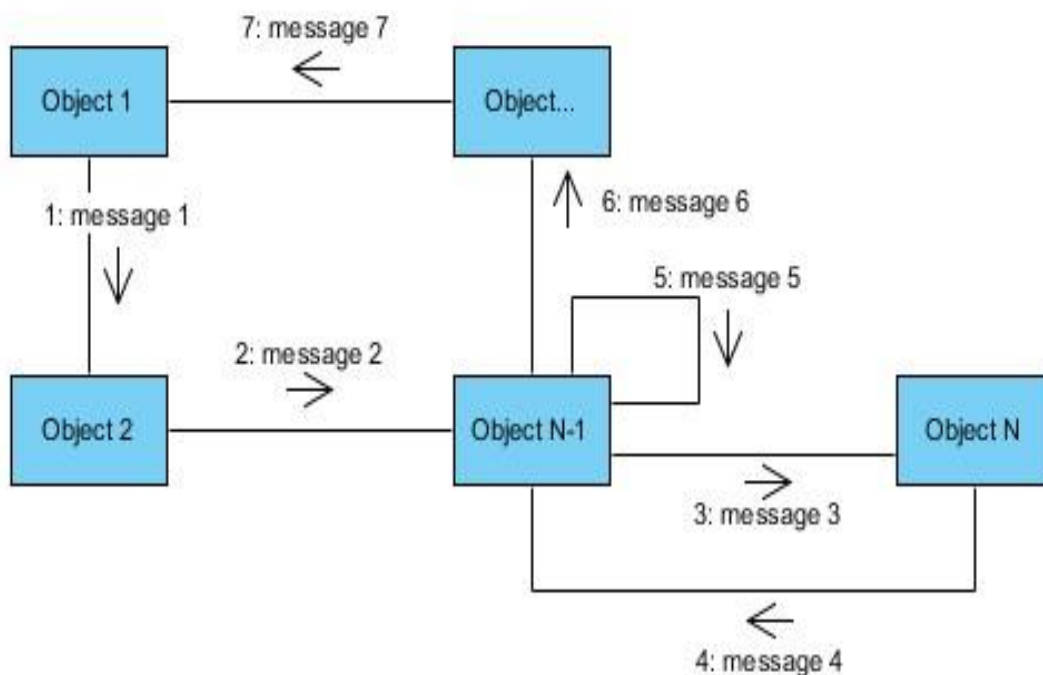
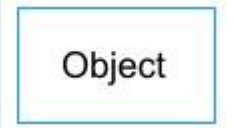



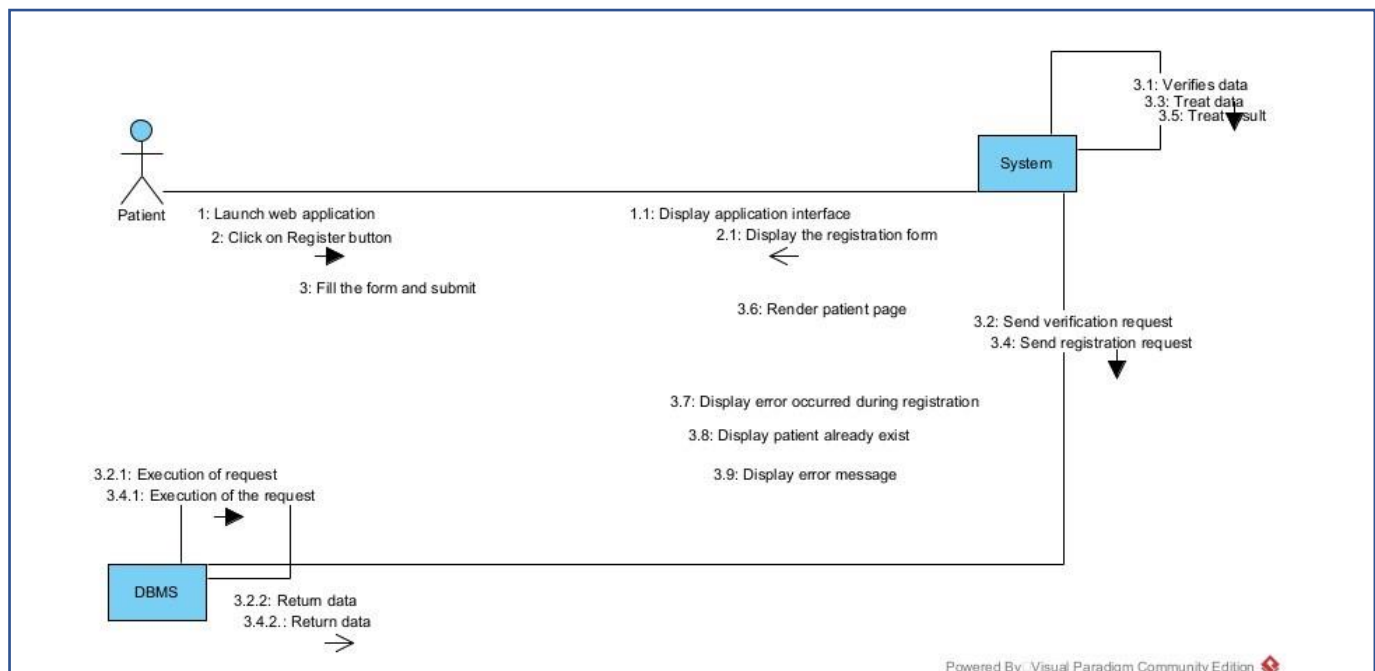
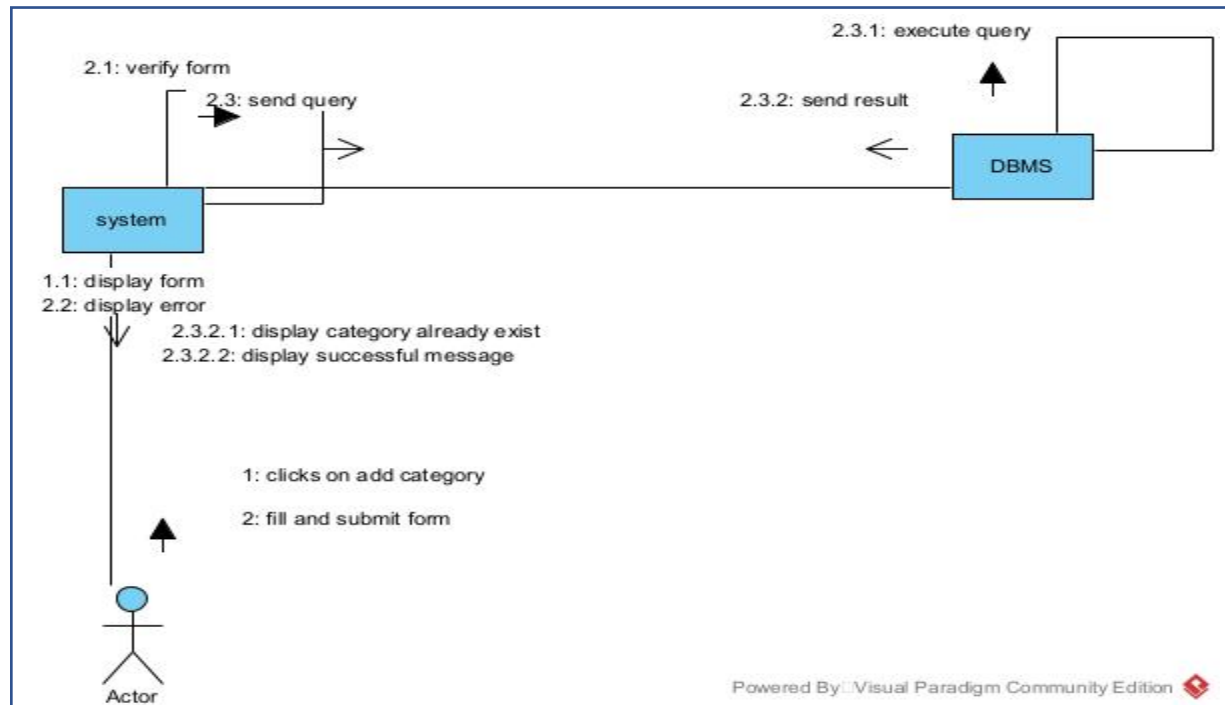


Figure 6 Communication diagram formalism



Table 8 Components of a communication diagram

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.



2. Sequence Diagram

a. Definition

A sequence diagram is a form of interaction diagram, which shows objects as lifelines running down the page and with their interactions over time represented as messages drawn as arrows from the source lifeline to the target lifeline. Sequence diagrams are good at showing, which objects communicate with which other objects and what messages trigger those communications. Sequence diagrams are not intended for showing complex procedural logic.

b. Formalism

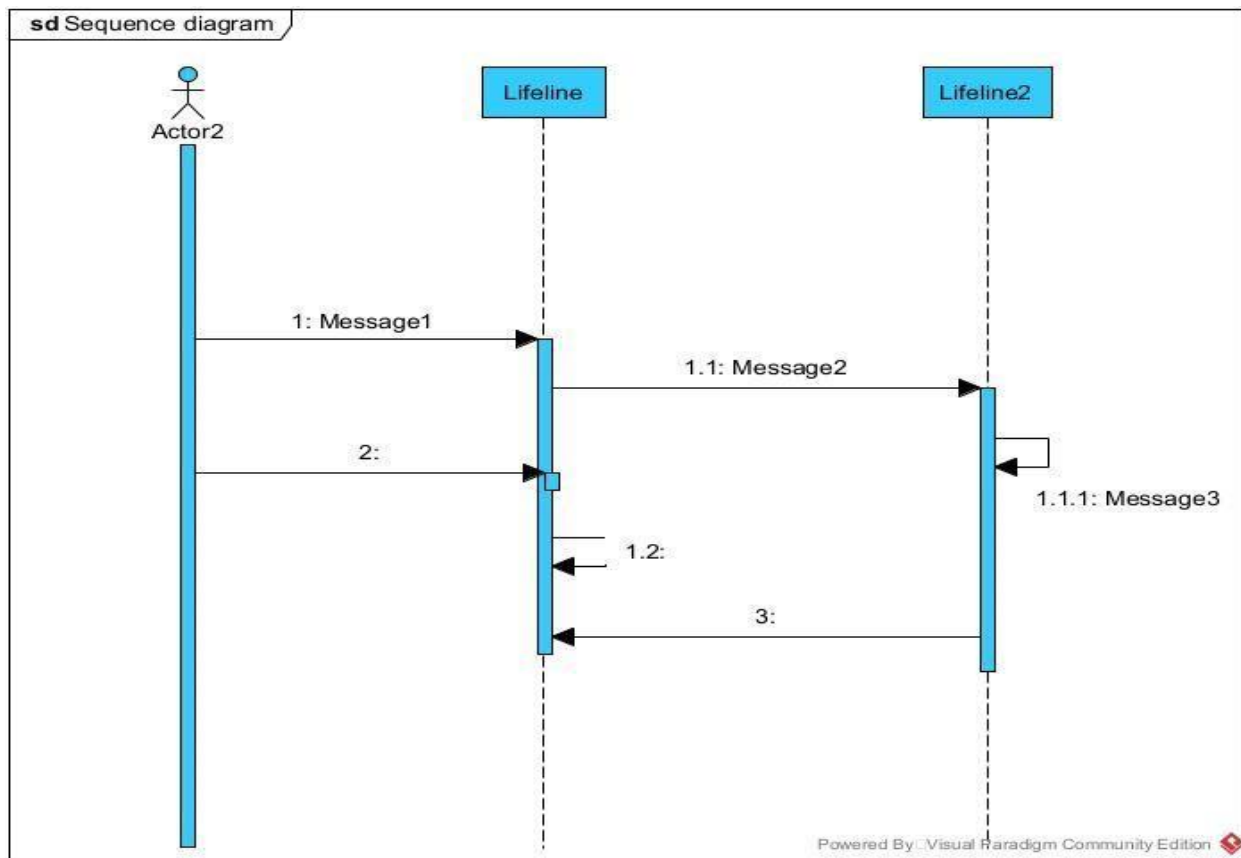
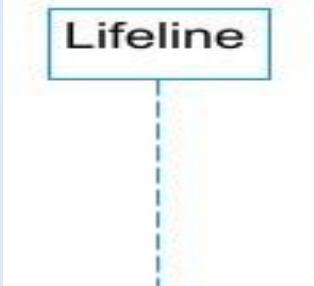
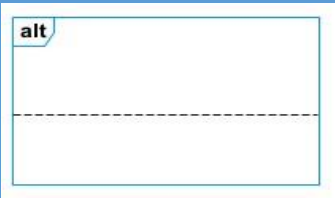

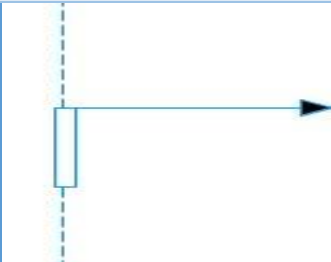


Figure 10 Formalism of a sequence diagram



c. Component of a sequence diagram

Table 9 Component of a sequence diagram

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 self messages.
Activation		It describes the time period in which an operation is performed by an element.

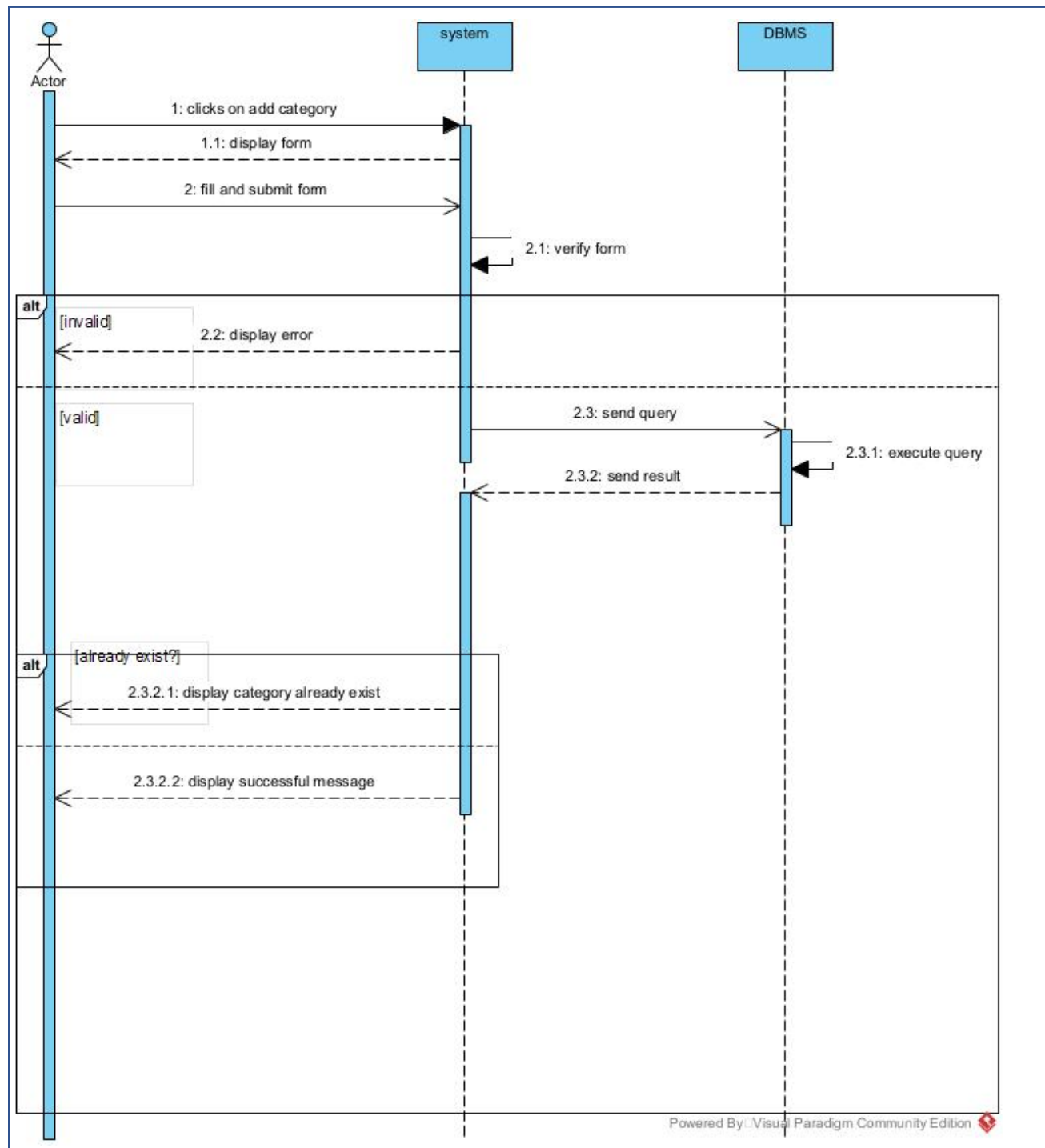
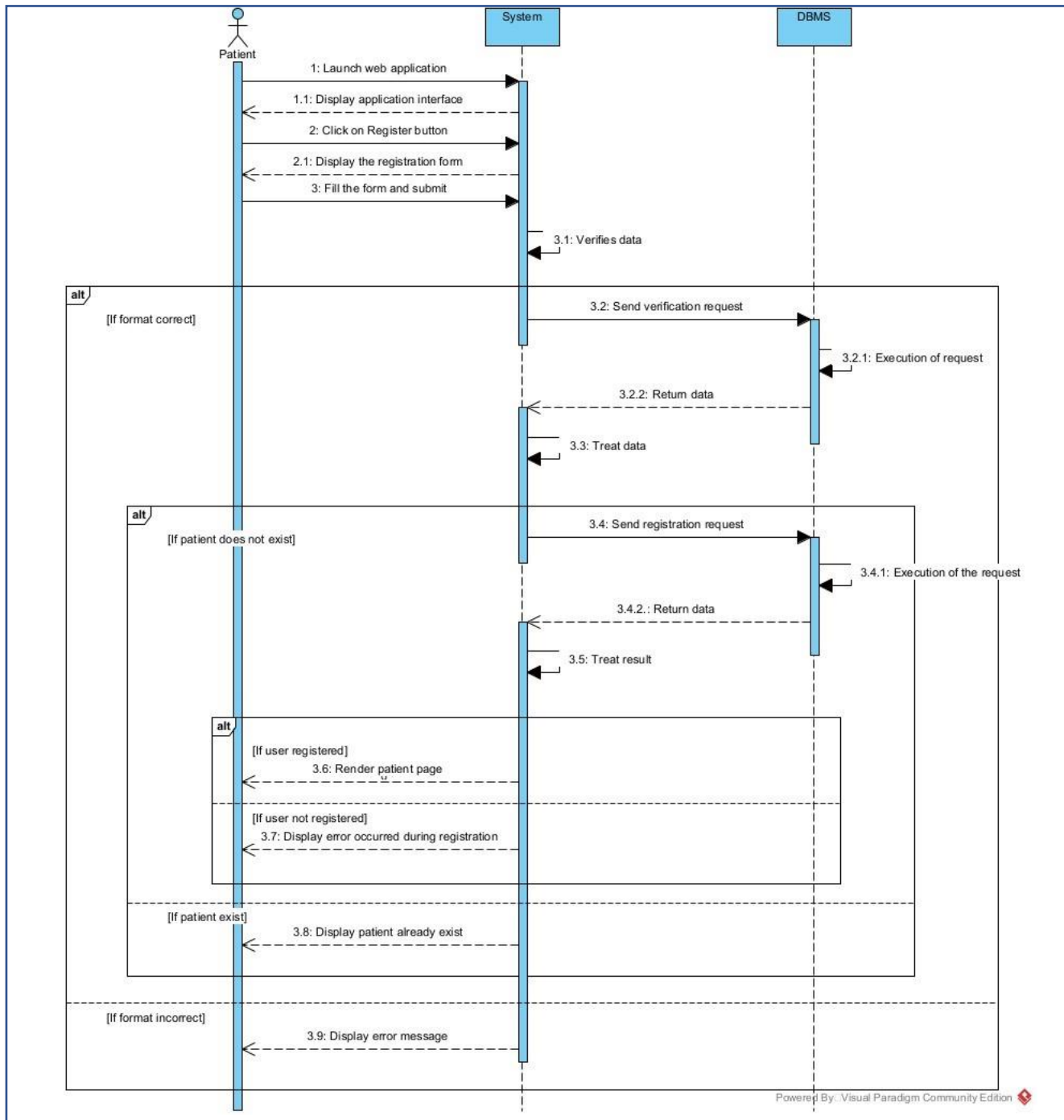


Figure 11 Sequence diagram add category



3. Activity diagrams

a. Definition

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

b. Formalism

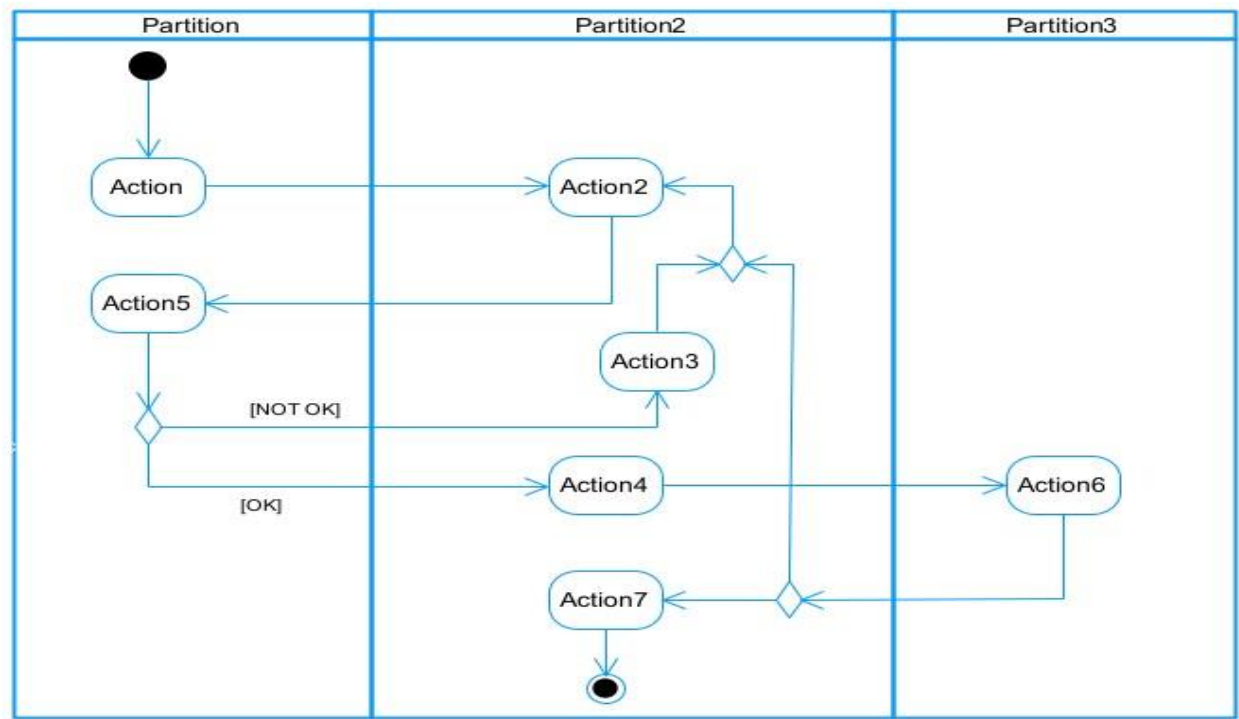
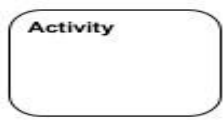

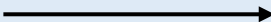


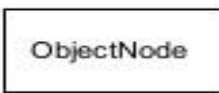
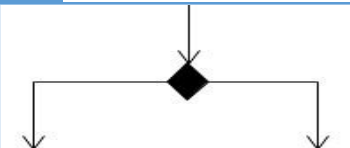


Figure 13 Formalism of an activity diagram

c. Components of an activity diagram

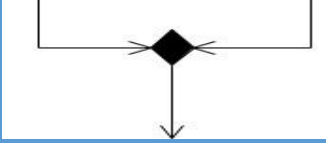
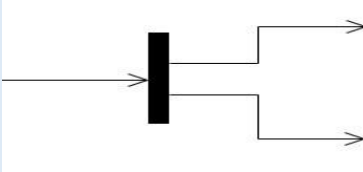
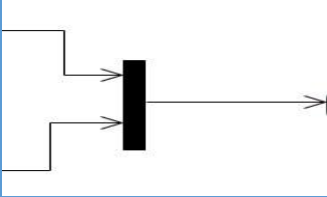
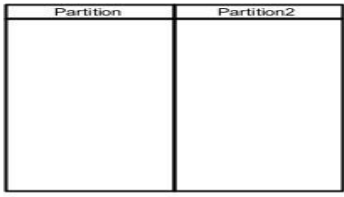
Table 10 Components of an activity diagram

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 splits an incoming activity edge into opposite outgoing activity edges.



DEVELOPMENT OF AN EXPENSE TRACKER APP



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.

d. Authentication activity diagram

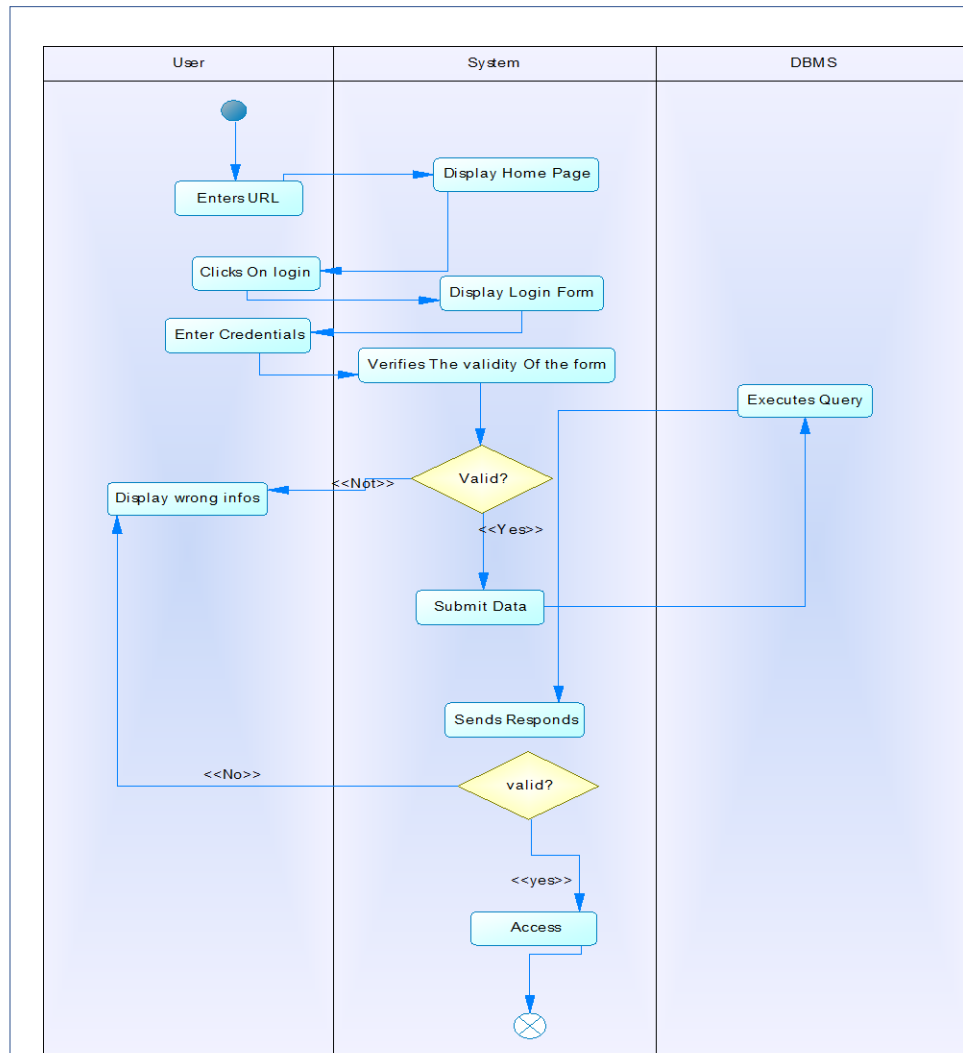


Figure 14 Authentication activity diagram

e. Activity diagram add category

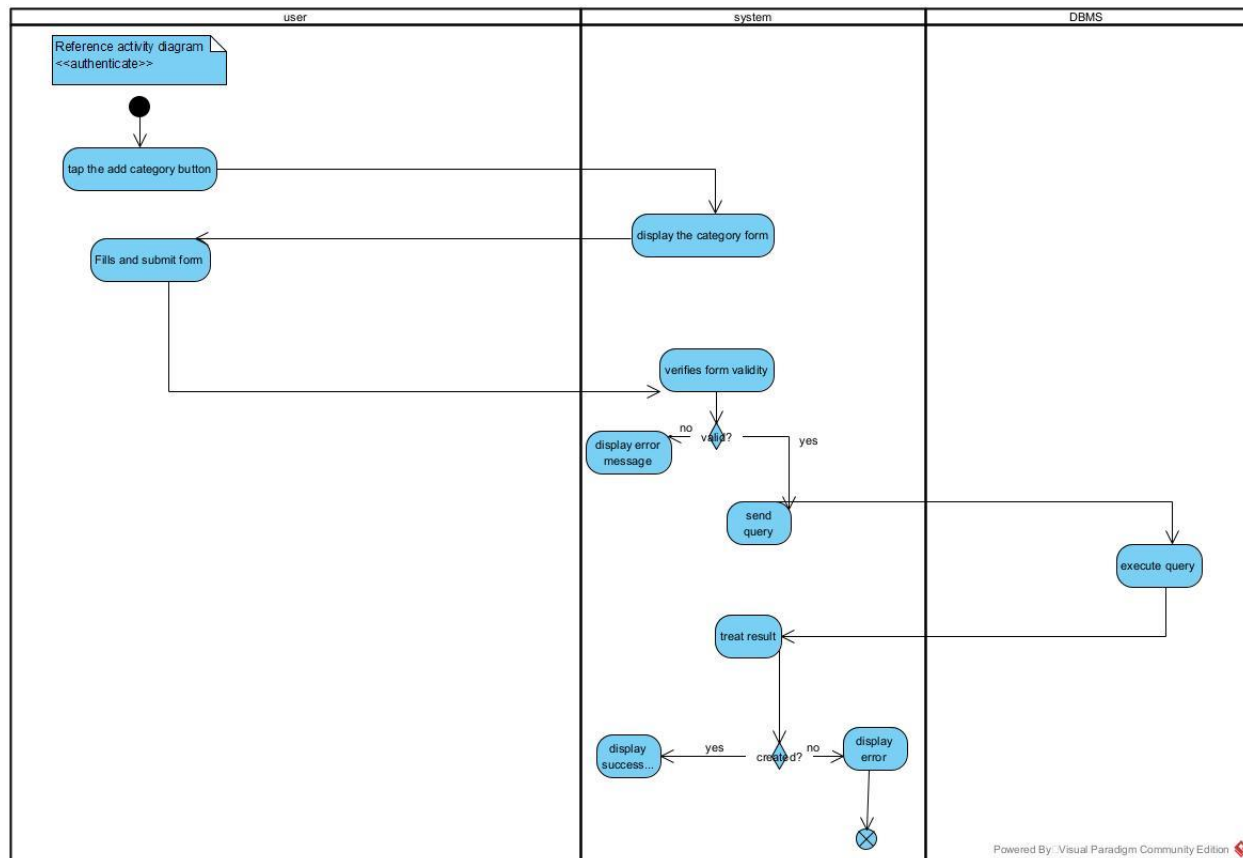


Figure 15 activity diagram add category



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 workflow 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.



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 bridges the gap between the analysis phase and the realization phase.

Content Overview

INTRODUCTION
I TECHNICAL BRANCH
II. IMPLEMENTATION BRANCH
CONCLUSION



INTRODUCTION

The conceptual phase will describe in details the necessary specifications, features and operations that will satisfy the functioning requirements of the proposed system as modelled 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 its objectives. We will look at some diagrams such as the component diagram, package diagram, communication diagram, class diagram and the object diagram.



I. TECHNICAL BRANCH

A. Generic Design

The generic conception consists of developing a solution that response to the technical specification. This conception is independent of the functional aspect specified in the functional branch of the 2TUP.

a. Physical architecture

The physical structure layer describes the system's hardware, software, and network environment. The physical architecture is based on the non-functional requirements such as availability, performance, security. The physical architecture of our system follows the 3-tier client-server architecture. Three-tier architecture is a client –server software architecture pattern in which the user interface (presentation), functional process logic (“business rules”), computer data storage and data access are developed and maintained as independent modules, most often on separate standalone application.

3-tier architecture has the following layers.

- Presentation layer (your PC, Tablet, mobile, etc.)
- Application layer (server)
- Database Server

b. Logical Architecture

Django is based on MVT (Model-View-Template) architecture. MVT is a software design pattern for developing a web application. MVT structure has the following three parts;

- ✚ **Model:** The model is going to act as the interface of your data. It is responsible for maintaining data. It is the logical data structure behind the entire application and it is represented by a database (generally relational database such as MySQL, Postgres)
- ✚ **View:** The view is the user interface, what you see in your browser when you render a website. It is represented by HTML/CSS/Javascript.
- ✚ **Template:** A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted.

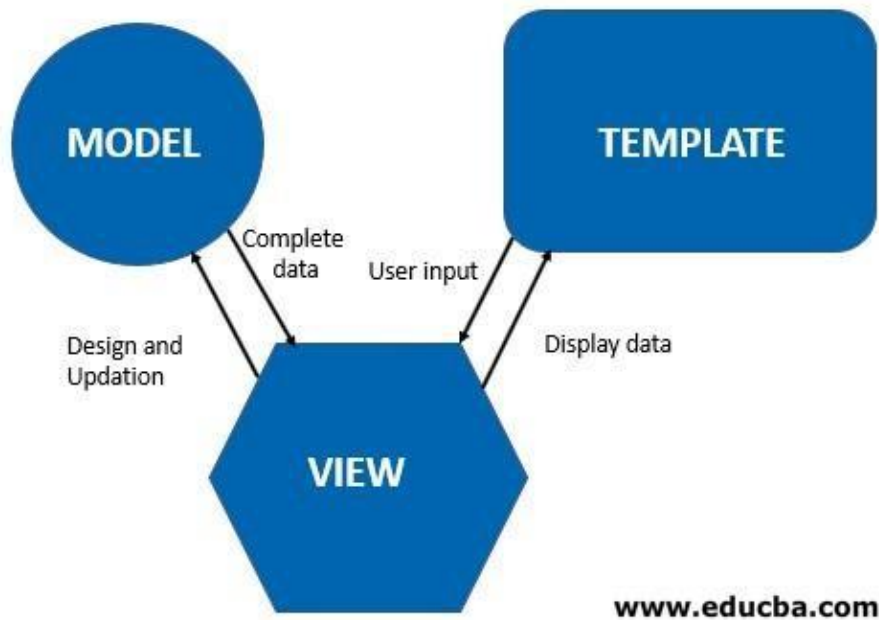


Figure 23: MVT architecture (source: <https://www.google.com/search?q=MVT+architecture>)



II. IMPLEMENTATION BRANCH

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

A. Preliminary Design

1. Class diagram

a. Definition

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

b. Formalism

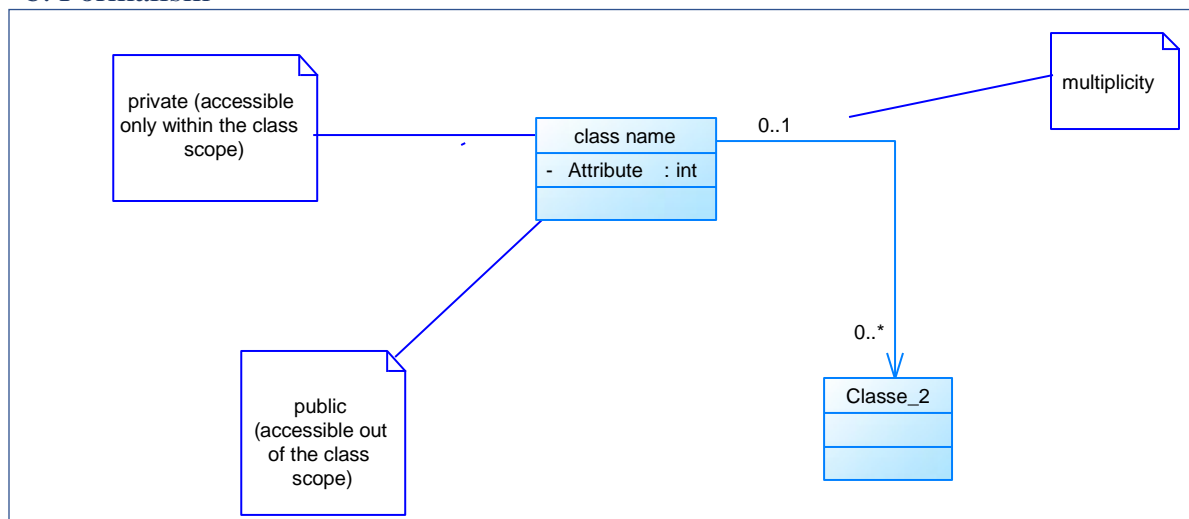


Figure 16 Formalism of a class diagram

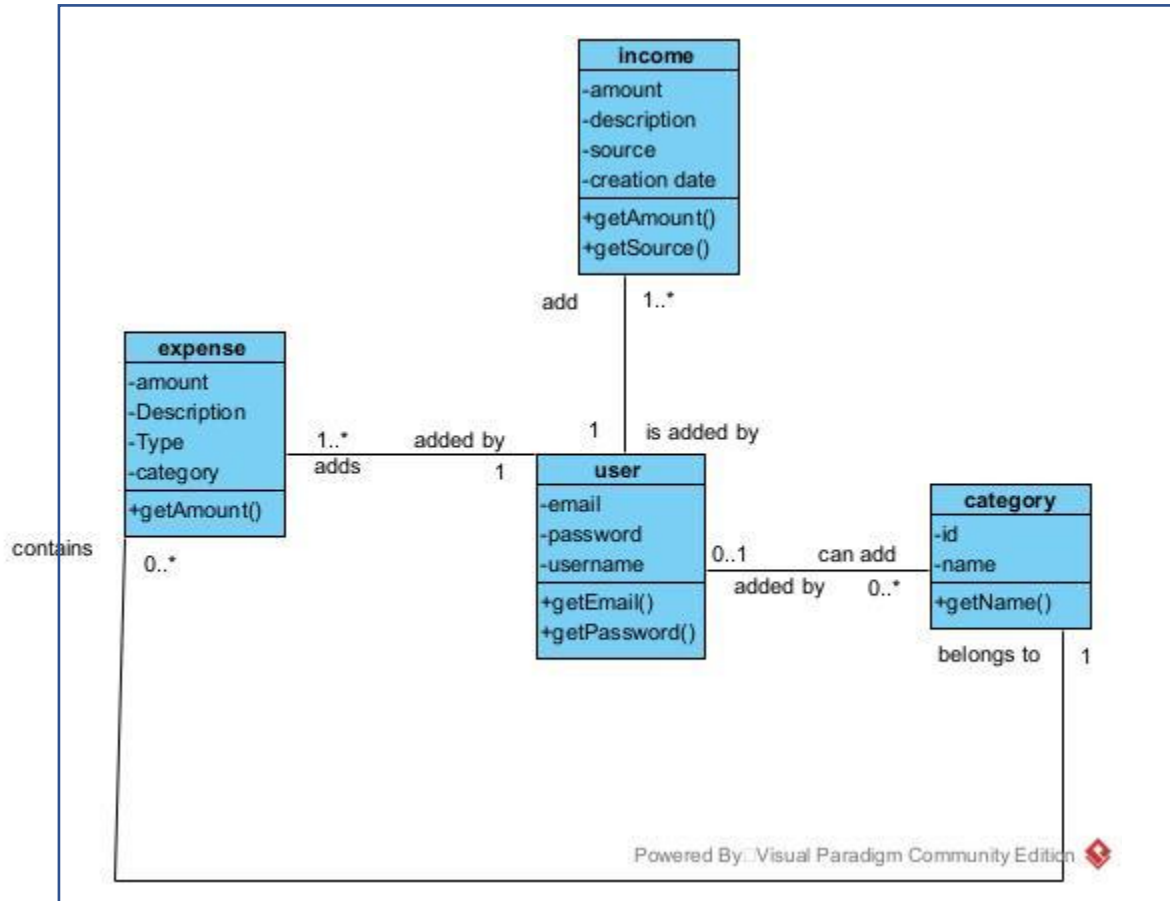


Figure 17 class diagram of spendwise

2. State machine diagram

a. Definition

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

b. Formalism

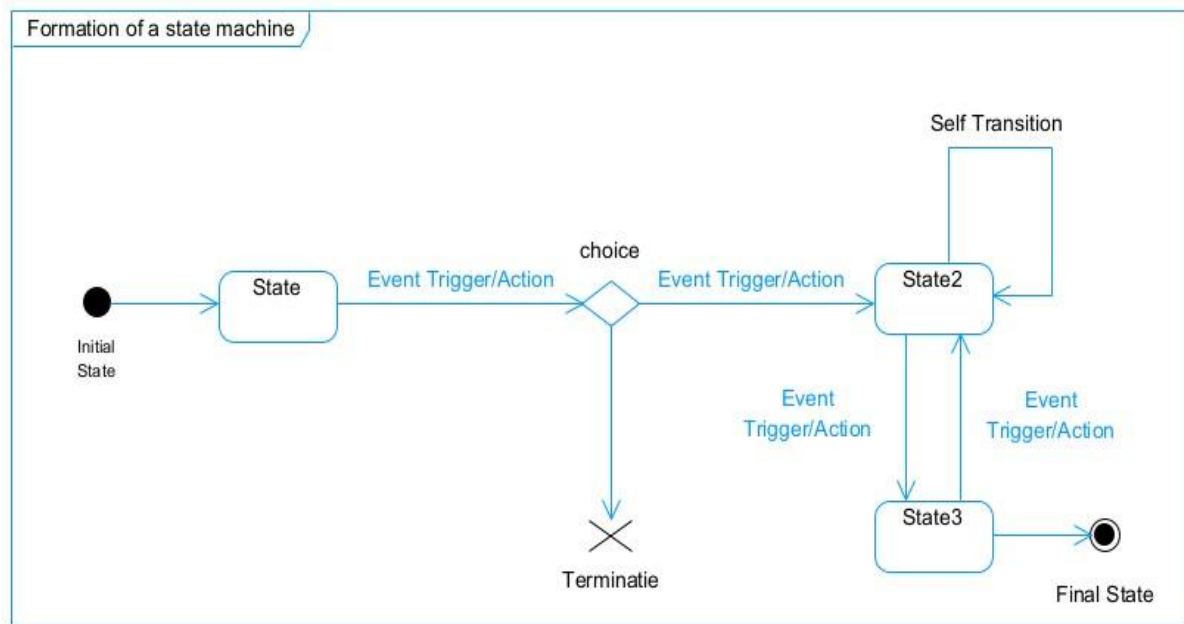
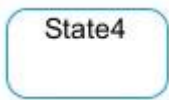





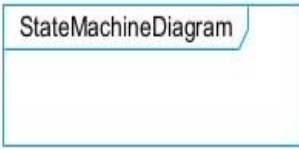


Figure 18 Class diagram formalism



d. Component of a class diagram

Element	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 its context is terminated.
Diagram Overview		A placeholder for the linked states in a state machine diagram.

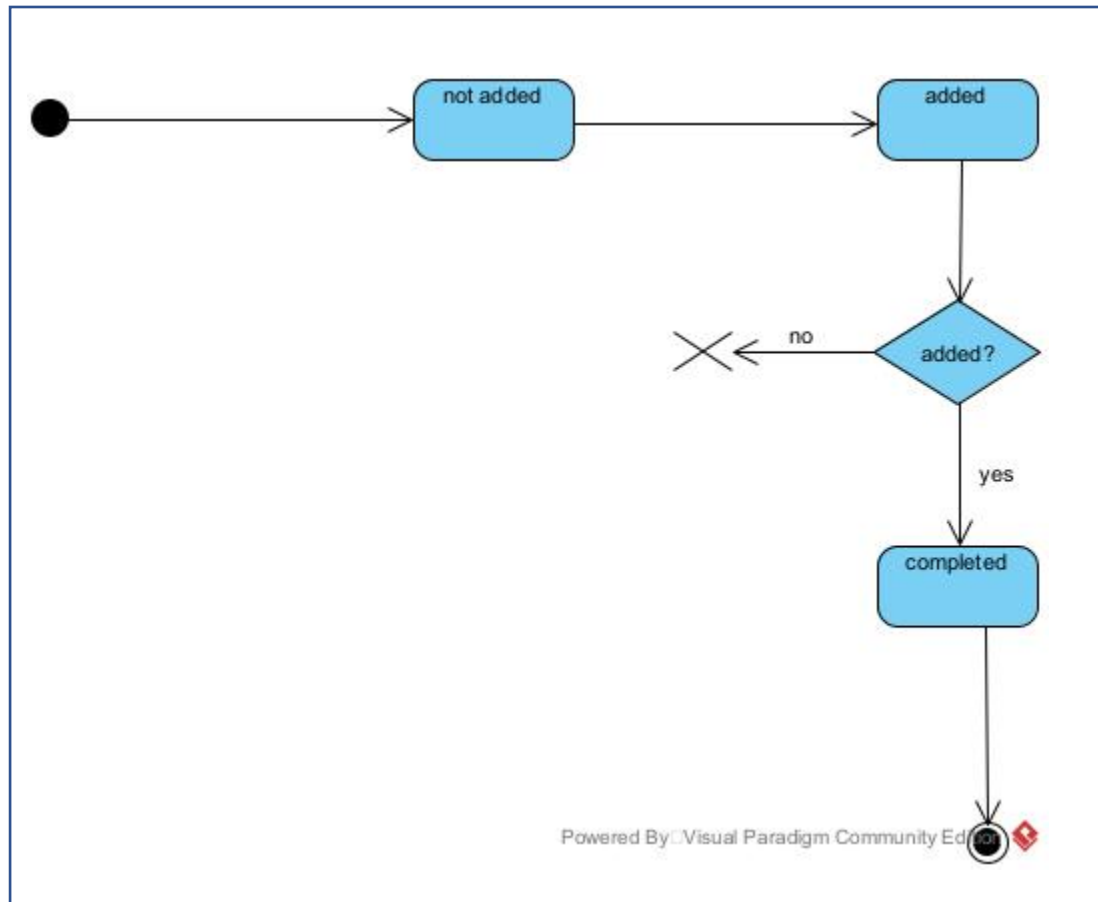


Figure 19 state machine diagram for add category

1. Package diagram

a. Definition

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

b. Formalism

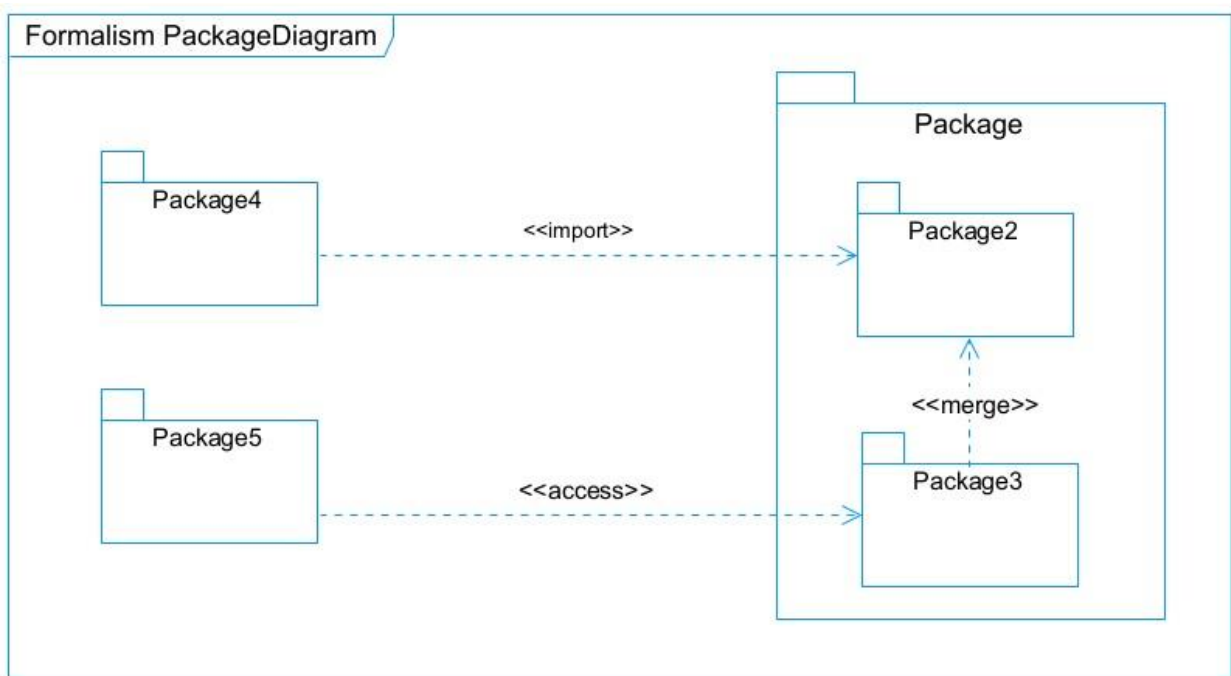
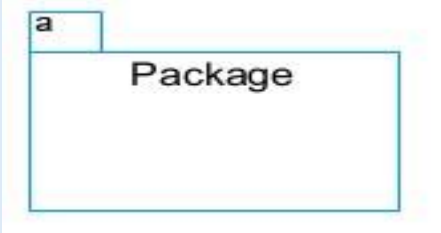
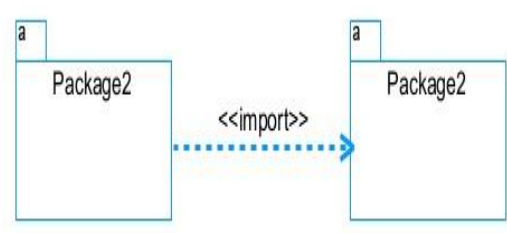
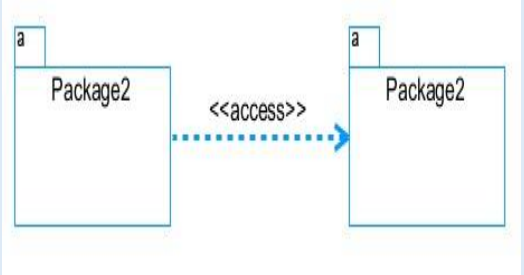


Figure 20 Package diagram formalism



c. Components of a package diagram

Table 11 components of a package diagram

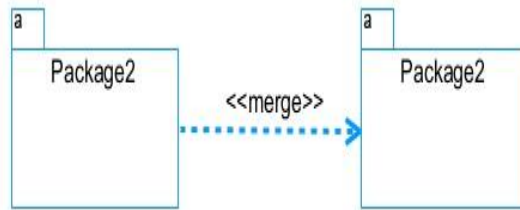
NAME	Representation	Description
Package		A package is a namespace use to group related elements; it is a mechanism used to group elements into a better structure in a system.
Package import		A relationship Indicate that, functionality has been imported from one package to another.
Package access		A relationship Indicates that one package requires assistance from the function of another package.



DEVELOPMENT OF AN EXPENSE TRACKER APP



Package merge



It is a relationship which shows that, the functionality of two packages are combined to a single function.



DEVELOPMENT OF AN EXPENSE TRACKER APP

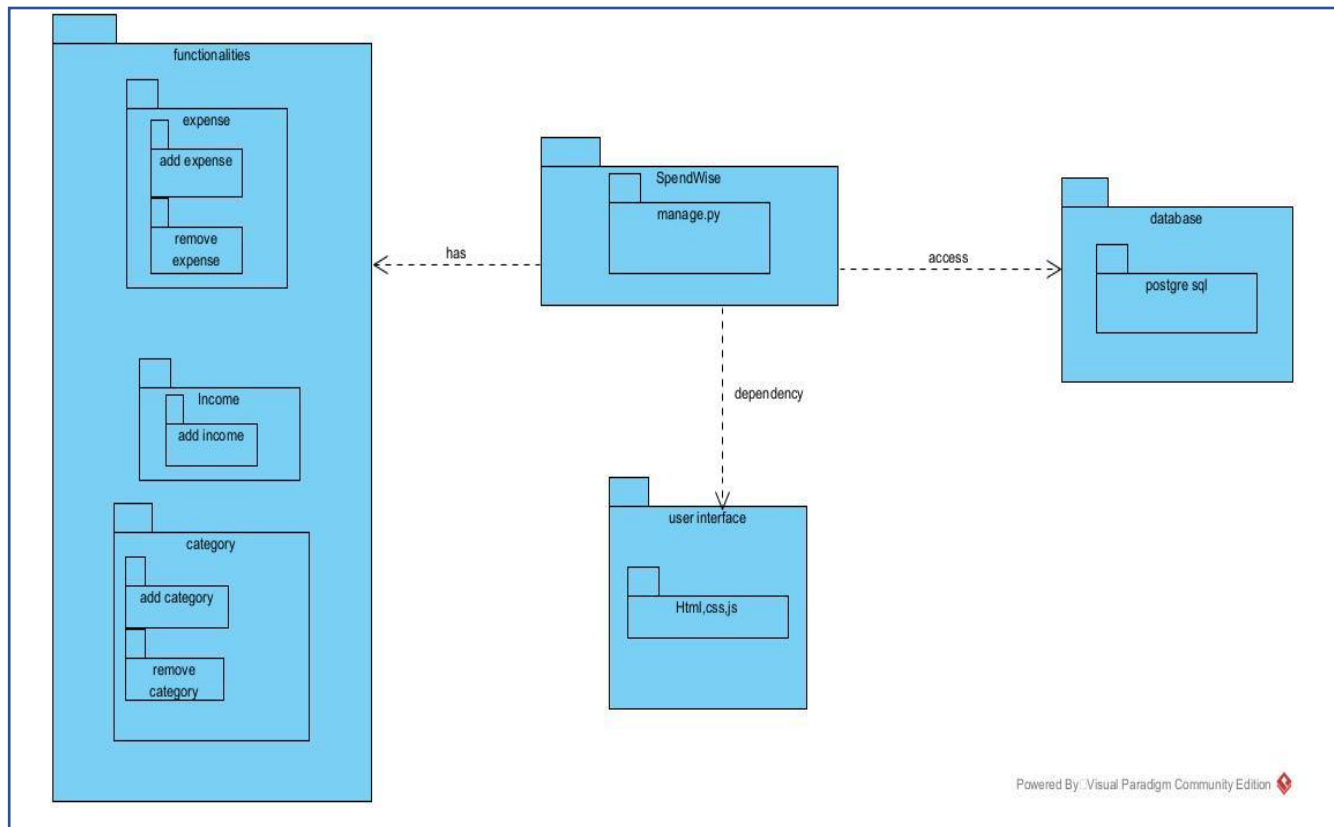


Figure 21 spend wise package diagram



PART SIX: REALIZATION PHASE



Preamble

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

INTRODUCTION

I. THE ENTITY RELATIONAL DIAGRAM

II. PRESENTATION OF THE DEVELOPMENT TOOLS

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.



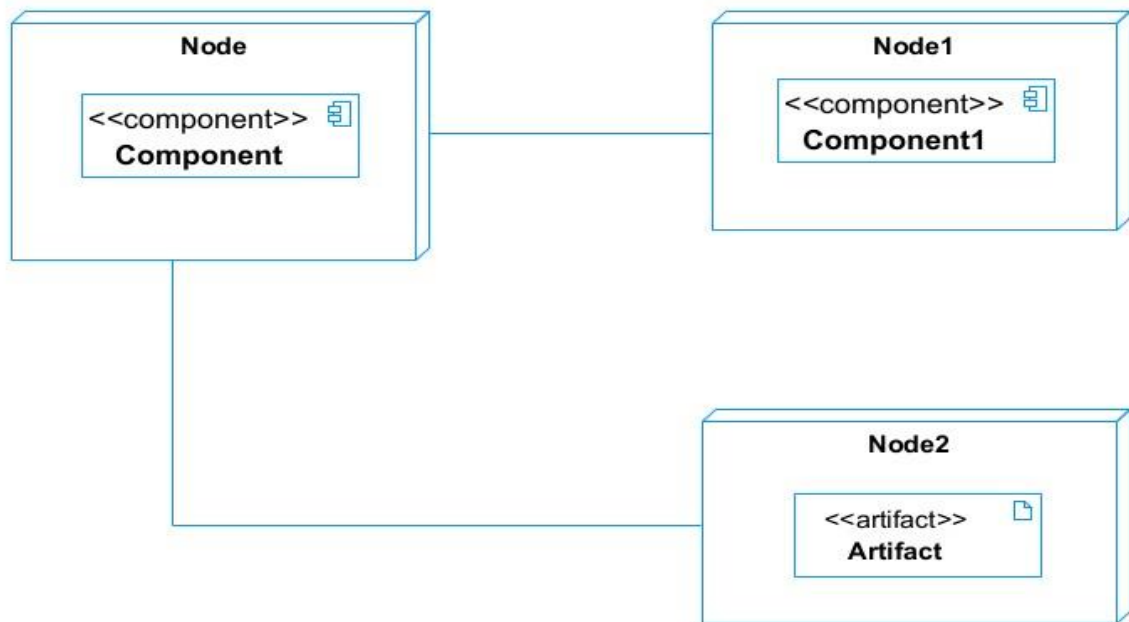
I. ENTITY RELATIONAL DIAGRAMS

1. Deployment Diagram

a. Definition

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

b. Formalism



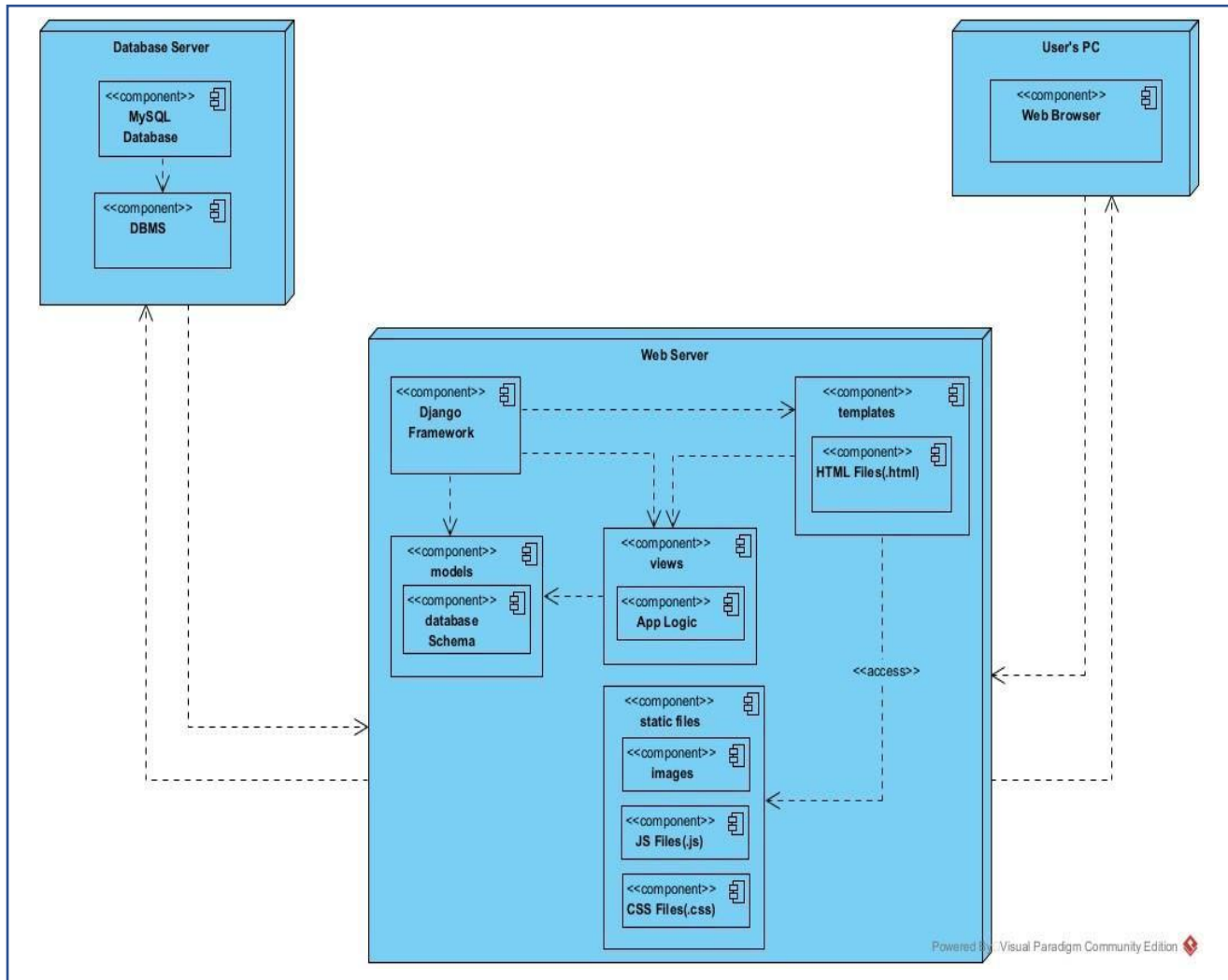


Figure 22 deployment diagram of spendwise

2. Component diagram

a. Definition

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

b. Formalism

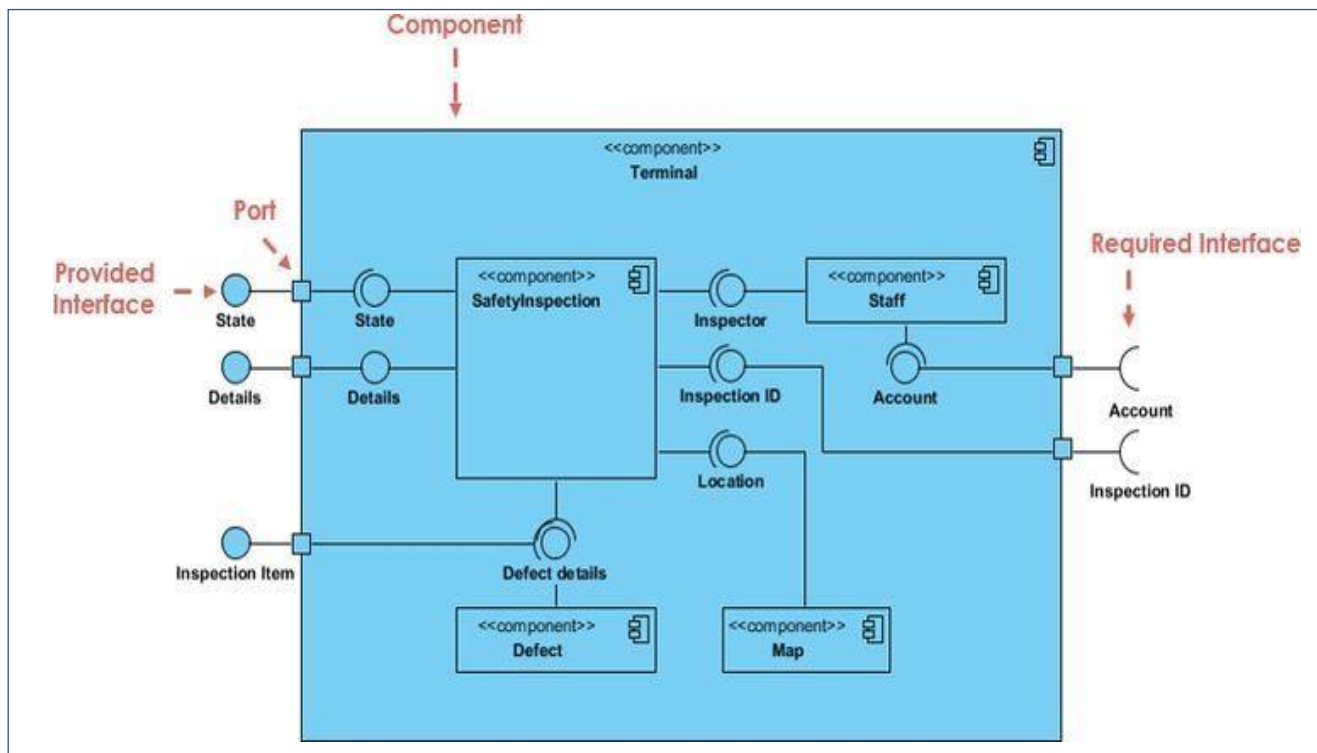


Figure 23 Formalism of a component diagram



c. Components of a component diagram

Table 12 Component 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 a smaller rectangle in the upper right corner which serves as its 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.

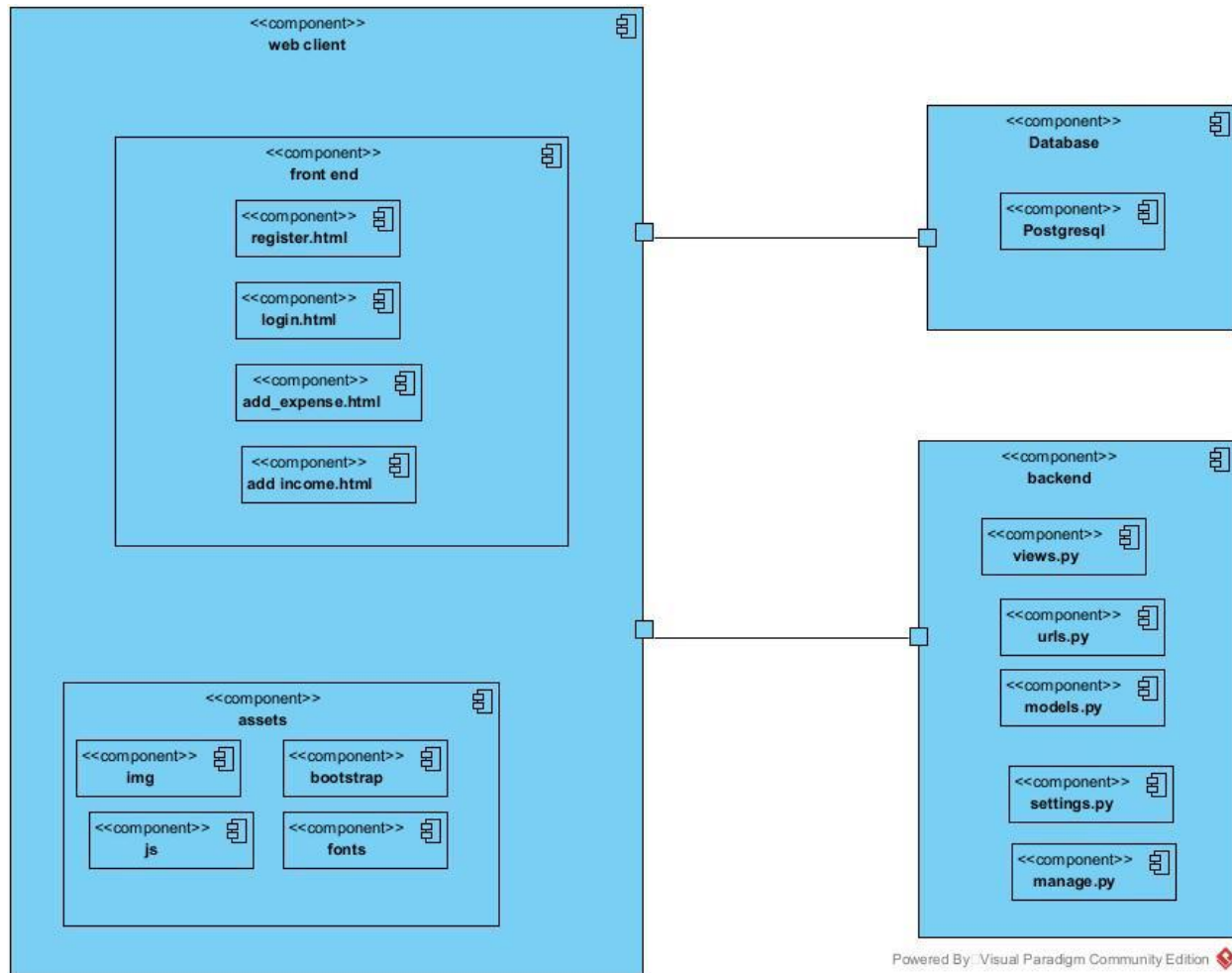


Figure 24 spendWise component diagram



II. PRESENTATION OF DEVELOPMENT TOOLS

A. Material resources

A development tool is a hardware or a software that supports the rapid implementation of software application. That is used to create, debug, maintain, or otherwise support other programs and applications.

B. TECHNOLOGICAL STACKS

Resource	Version	License Owner	Function
Microsoft Visual Studio Code	1.60	Microsoft Corporation	Text editing and coding



CONCLUSION

In our realization phase, we implemented our application. In other to accomplish this phase, we made used of our analysis and conception phase. The Entity-Relational diagram. helped us to establish our database. We then presented the development tools we used for the development or realization of our system, we had material resource (visual studio code) and technology. We will move to the user guide phase where we shall elaborate on how to install and use **SpendWise**.



PART SEVEN: 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

I. DEPLOYMENT OF THE WEB APPLICATION

II. SHOW CASE

CONCLUSION



INTRODUCTION

This is the final phase of our report. In this phase we will walk through the requirements for our system, the necessary installation process, accessing our system and its futures, all this in a step-by-step manner to facilitate the setting up of the platform for the first time users. The steps of different processes will be accompanied by images. After we will showcase our application by viewing the different screens and some brief explanation.



DEPLOYMENT OF THE WEB APPLICATION

I. INSTALLATION

STEP 1: Downloading the PyCharm installer package

We go to <https://www.jetbrains.com/pycharm/download> in order to download PyCharm and note that we are downloading the community version since it is free and open source.

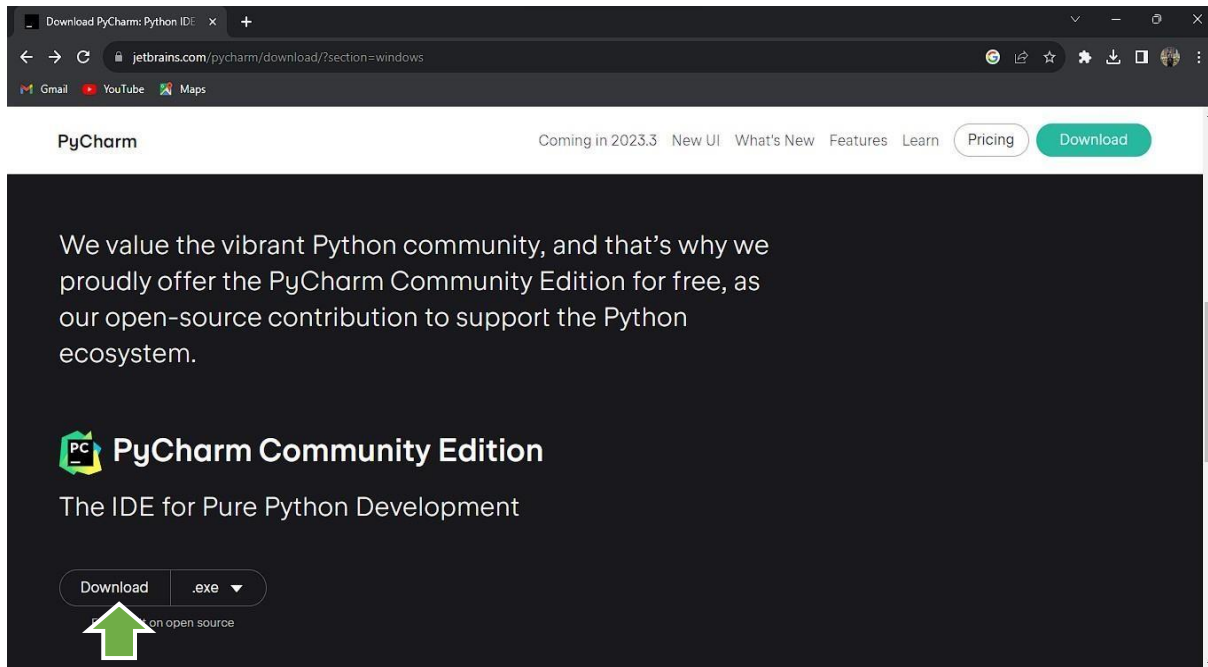


Figure 25 pycharm installation

STEP 2: Install PyCharm with the installation wizard

- A. Navigate to your downloads folder and double click on the .exe package you just downloaded.



DEVELOPMENT OF AN EXPENSE TRACKER APP

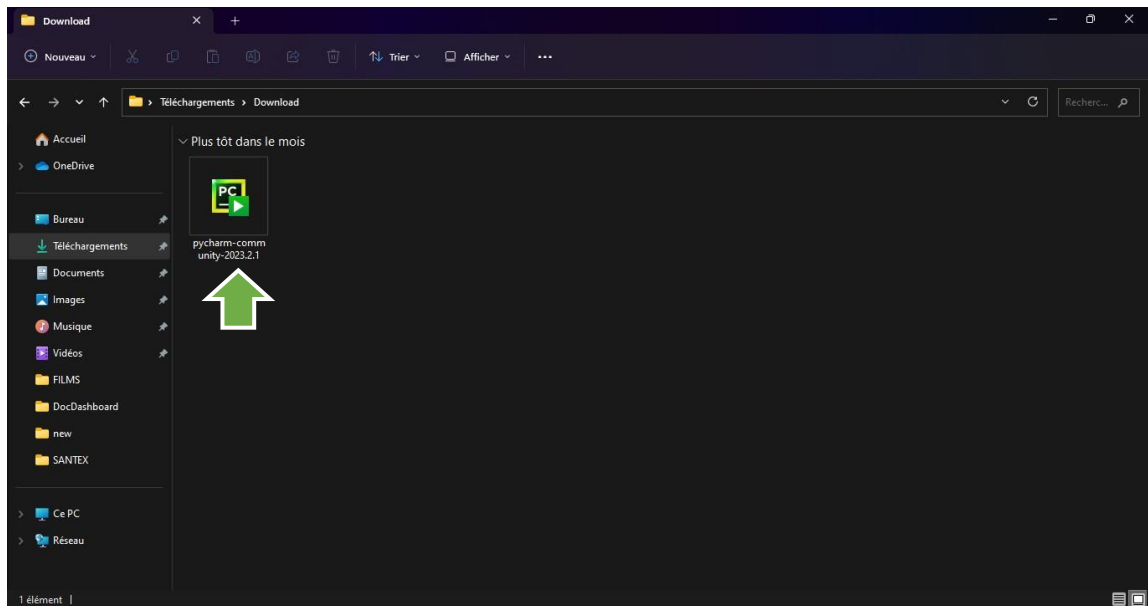


Figure 26 : Launching the installation wizard

Click on next button

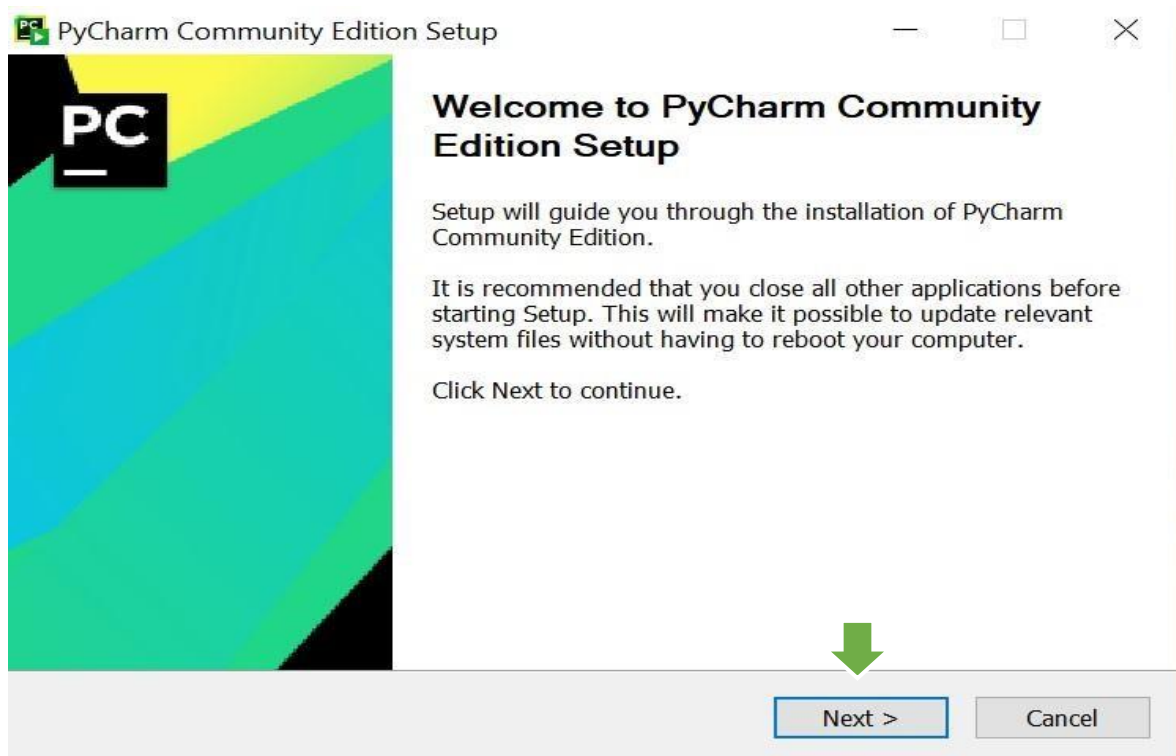


Figure 27 Clicking on next



F. Click on finish to complete the installation

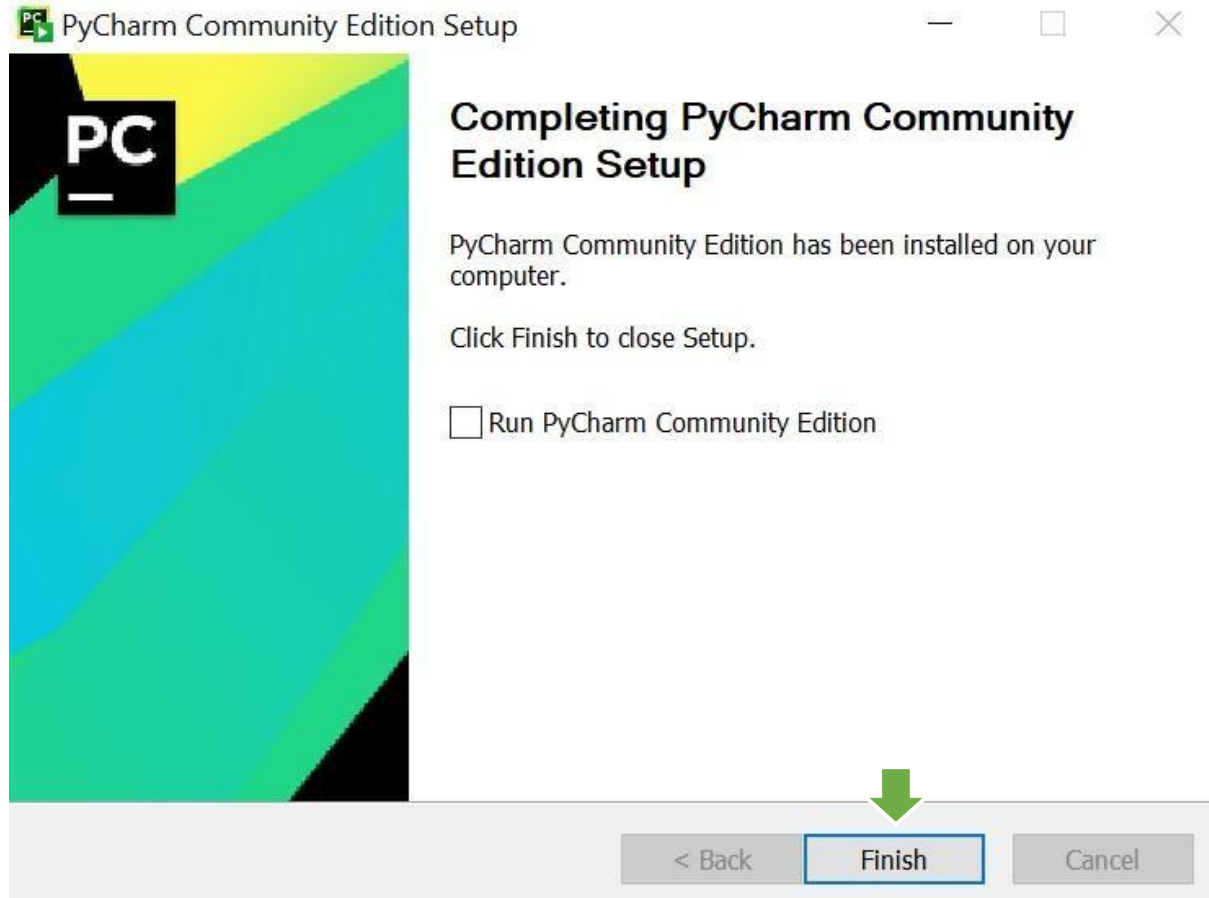
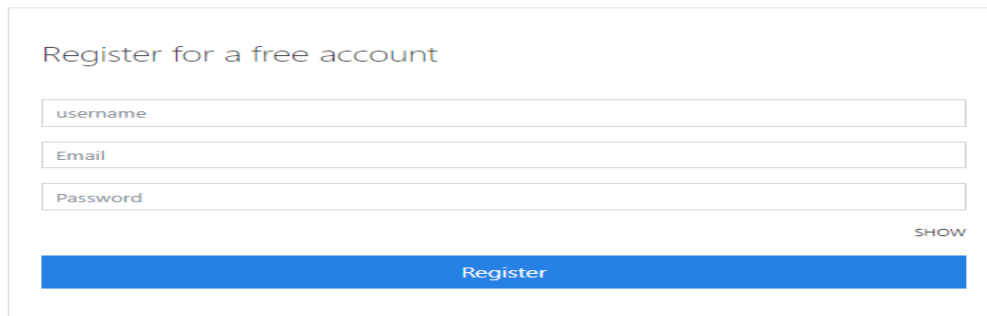


Figure 28 finish



II. Show case

a. Register Form



Register for a free account

username

Email

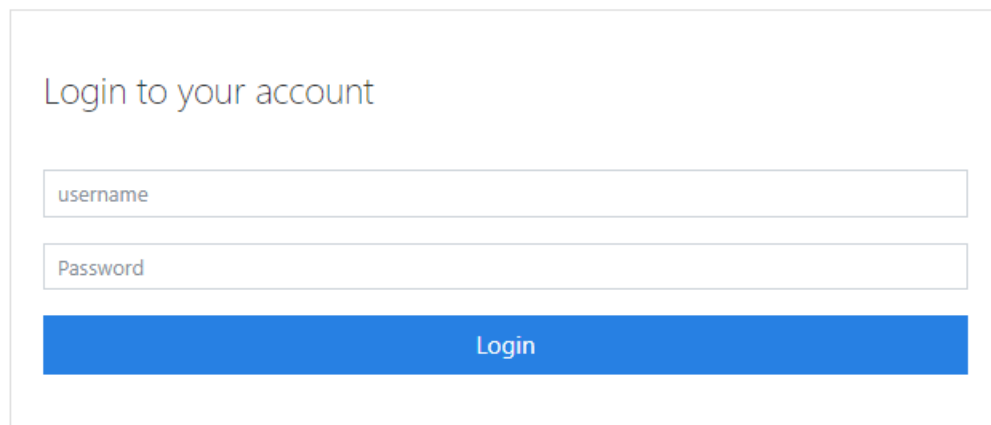
Password

SHOW

Register

Figure 29 Register form

b. Login form



Login to your account

username

Password

Login

Figure 30 login form



c. User dashboard

TrulyExpenses

(wemayi)

Sign Out

Dashboard

Expenses

Income

SUMMARY

Expenses Summary

Income Summary

SETTINGS

General

Account

Expenses / My Expenses

Add Expense

Search

Amount (None)	Category	Description	Date	
200.0	Groceries	bread	July 14, 2024	Edit

Showing page 1 of 1

Figure 31 dashboard

d. Admin page:

TrulyExpenses	
Site administration	
AUTHENTICATION AND AUTHORIZATION	
Groups	+ Add Change
Users	+ Add Change
EXPENSES	
Categories	+ Add Change
Expenses	+ Add Change
USERINCOME	
Sources	+ Add Change
User incomes	+ Add Change

Recent actions

My actions

None available

Figure 32 admin page



CONCLUSION

Having put in place the application, it was not sufficient we had to produce a manual that will helps its various users. That is why we presented the way the application was deployed and all the functionalities of this application. It also guides the users on how to use the application.



PERSPECTIVES

Considering the fact that this project can be improved to a higher extend, as perspectives we can bring some ameliorations such as:

- ✚ The association of an android application to the platform.
- ✚ The possibility to implement an AI that will permit to predict future expenses and give advice to the client.



GENERAL CONCLUSION

Having come to an end of our project, we have achieved a lot from our internship period, it was very challenging and demanding, we had to manage stress, failures and successes.

The theme “EXPENSE TRACKER”, our greatest desire is to facilitate expense tracking. We began by identifying the current system in place for the consultation process, project constraints and requirements in the specification book, then we went forth to analyze our system (SpendWise) using UML-2TUP methodology. We made use of Sybase Power AMC, a modelling software used to draw our various diagrams. In order to realize our project, we used HTML, CSS, JS for the front-end, Django for the backend and PGSQL as our database. We won't end here, updates and important improvements are going to be made in the nearest future in terms of functionality, security to make it more reliable.



ANNEXES



BIBLIOGRAPHY

- † Mercurial 2023
- † Django documentation
- † UML notes by Mrs. Tchinga Alice



WEBOGRAPHY

- † <https://www.tutorialspoint.com/uml/index.htm> ; for learning more about UML diagrams
- † <https://docs.djangoproject.com/>; for learning more about Django
- † <https://www.tutorialspoint.com/bootstrap/index.htm> ; to learn how to code the front end of the application with bootstrap
- † <https://www.google.com/search?q=tuto+on+2tup&oq=tuto+on+2tup>; to know more about 2TUP process
- † <https://www.youtube.com/watch?v=mQnru1TzWks>; learn to design the frontend



TABLE OF CONTENT

Contents

DEDICATION	ii
ACKNOWLEDGMENTS	iii
SUMMARY	iv
LIST OF ABBREVIATIONS	v
LIST OF FIGURES	vi
LIST OF TABLES	viii
ABSTRACT	ix
RESUME	x
GENERAL INTRODUCTION	1
PART ONE: EXISTING SYSTEM	2
INTRODUCTION	4
I. THEME PRESENTATION	5
II. STUDY OF THE EXISTING SYSTEM	6
III. CRITICISM OF THE EXISTING SYSTEM	7
IV. PROBLEMATIC	8
V. PROPOSED SOLUTION	9
CONCLUSION	10
PART TWO: SPECIFICATION BOOK	11
INTRODUCTION	13
I. CONTEXT AND JUSTIFICATION	14



II.	OBJECTIVES OF THE PROJECT.....	15
A.	General objective of the project.....	15
B.	Specific objectives	15
III.	EXPRESSION OF NEEDS.....	16
A.	Functional needs	16
B.	Non-functional Needs	17
IV.	ESTIMATED COST OF THE PROJECT	18
A.	Hardware Resources	18
B.	Software Resources	18
C.	Human Resources	20
D.	Total Project Estimated Cost	20
A.	List of participants	21
B.	Deliverables	21
	CONCLUSION.....	22
	PART FOUR: ANALYSIS PHASE	23
	INTRODUCTION	25
II.	PRESENTATION OF THE ANALYSIS METHOD	26
A.	MODELING WITH UML 2.5.....	27
B.	UML 2.5 diagrams overview.....	29
C.	The Process of Development of a Software	30
D.	A Unified Process	30
E.	The Two Track Unified Process (2TUP).....	30
III.	JUSTIFICATION OF THE ANALYSIS METHOD.....	33



IV. MODELLING OF THE PROPOSED SOLUTION	34
A. Capture of the functional needs	34
CONCLUSION	53
PART FIVE: CONCEPTION PHASE	54
INTRODUCTION	56
I. TECHNICAL BRANCH	57
A. Generic Design	57
b. Logical Architecture	58
II. IMPLEMENTATION BRANCH	59
A. Preliminary Design	59
PART SIX: REALIZATION PHASE	68
INTRODUCTION	70
I. ENTITY RELATIONAL DIAGRAMS	71
II. PRESENTATION OF DEVELOPMENT TOOLS	76
A. Material resources	76
B. TECHNOLOGICAL STACKS	76
PART SEVEN: USER GUIDE	78
INTRODUCTION	80
DEPLOYMENT OF THE WEB APPLICATION	81
I. INSTALLATION	81
STEP 1: Downloading the PyCharm installer package	81
STEP 2: Install PyCharm with the installation wizard	81
II. Show case	84



DEVELOPMENT OF AN EXPENSE TRACKER APP



CONCLUSION.....	86
PERSPECTIVES	87
GENERAL CONCLUSION	88
BIBLIOGRAPHY	XII
WEBOGRAPHY	XIII
TABLE OF CONTENT	XIV