THIS DOCUMENT IS DEDICATED TO THE BALEMBA’S FAMILY

DEDICATION

ACKNOWLEDGEMENT

We owe a lot of thanks to some people who helped us make this document possible. They dedicated their time and effort to ensure the completion of this work. We would like to express our appreciation to the following people:

The Resident Representative of IAI-Cameroon, **Mr. Abanda ARMAND Claude**, for

his support, words of encouragement and the different advices on how to approach

situations.

➢ The chief executive officer of ICES SARL, **Mr. JOSEPH Kobe** and his staffs for

trusting us by given us the opportunity to serve as interns in their prestigious enterprise.

➢ Our professional supervisor, **Mr. DARYL Ngala** for her moral support, advice and professional expertise she shared with us.

➢ Our academic supervisor **Mr. JIONGANG THIBAUT**. for her moral support, effort and

advices throughout the academic year.

➢ To our academic teachers **Mr. AGBOR ANDERSON**, **Mrs. TCHINGA Alice**

AND **Mr. NDENGE** for their advices and assistance in realizing this document.

➢ To all my teachers of **AICS CAMEROON** for the knowledge they impacted me with

for me to be where I am today.

➢ For my brothers and sister for their love and support.

➢ The countless contributors of open-source programming community, for their great

help in learning basic skills and detecting and solving bugs.

➢ To **Mr. & Mrs. BALEMBA** for their moral and material support.

➢ To all my classmates for their collaborative work throughout the academic year.

[**Error! Bookmark not defined.**](#_Toc147485934)

[PART I 2](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147485935)

[INSERTION PHASE 2](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147485936)

[4](#_Toc147485937)

[PART II 19](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147485938)

[EXISTING SYSTEM 19](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147485939)

[PART III 27](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147485940)

[SPECIFICATION BOOK 27](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147485941)

[INTRODUCTION 28](#_Toc147485942)

[I. CONTEXT AND JUSTIFICATION OF STUDIES 29](#_Toc147485943)

[OBJECTIVES OF THE PROJECT 30](#_Toc147485944)

[II. PLANNING OF THE PROJECT 32](#_Toc147485945)

[III. ESTIMATION OF THE PROJECT 35](#_Toc147485946)

[IV. PROJECT CONSTRAINTS 37](#_Toc147485947)

[V. DELIVERABLES 38](#_Toc147485948)

[CONCLUSION 39](#_Toc147485949)

[PART IV 41](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147485950)

[ANALYSIS PHASE 41](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147485951)

[PART IV 80](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147485952)

[CONCEPTION PHASE 80](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147485953)

[PART V 100](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147485954)

[PART VI 113](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147485955)

[FUNCTIONALITY TEST 113](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147485956)

[PART VII 119](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147485957)

[USER GUIDE 119](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147485958)

[INTRODUCTION 121](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147485959)

SUMMARY

LIST OF FIGURES

[Figure 1: Company Infrastructure Services Provided by ICES\_SARL 8](#_Toc147485660)

[Figure 2: Geo Localisation of ICES\_SARL 10](#_Toc147485661)

[Figure 3: Organization Chart of ICES\_SARL 13](#_Toc147485662)

[Figure 4: Gantt projectt 34](#_Toc147485663)

[Figure 5:UML Diagrams Overview (Source: UML 2 For Dummies,2003 p.19) 48](#_Toc147485664)

[Figure 6:Overview of 2TUP(Source: https://cedric.cnam.fr/fichiers/RC967.pdf p.3) 52](#_Toc147485665)

[Figure 7:UseCase Diagram Formalism 54](#_Toc147485666)

[Figure 8:General Use Case Diagram 58](#_Toc147485667)

[Figure 9:Student Overview Use Case diagram 59](#_Toc147485668)

[Figure 10:Instructor Overview Use Case Diagram 60](#_Toc147485669)

[Figure 11:Admin Overview Use Case Diagram 61](#_Toc147485670)

[Figure 12: Communication Diagram Formalism 66](#_Toc147485671)

[Figure 13:System Login Communication diagram 68](#_Toc147485672)

[Figure 14: Sequence Diagram Formalism 69](#_Toc147485673)

[Figure 15:Sequence Diagram:SignUp 72](#_Toc147485674)

[Figure 16:Sequence diagram: Login 73](#_Toc147485675)

[Figure 17:Activity diagram formalism 74](#_Toc147485676)

[Figure 18: SignUp Activity Diagram 77](#_Toc147485677)

[Figure 19: Login Activity Diagram 78](#_Toc147485678)

[Figure 20:3-Tier architecture(Source: https://mobidev.biz/blog/web-application-architecture-types) 84](#_Toc147485679)

[Figure 21:MVC Architecture (Source: https://www.crio.do/blog/understand-mvc-architecture/) 85](#_Toc147485680)

[Figure 22:Hardware Diagram 86](#_Toc147485681)

[Figure 23:High Level Architecture 87](#_Toc147485682)

[Figure 24:State Machine Diagram formalism 88](#_Toc147485683)

[Figure 25:Login State Machine Diagram 90](#_Toc147485684)

[Figure 26:System SignUp State Machine Diagram 91](#_Toc147485685)

[Figure 27:Package Diagram Formalism 92](#_Toc147485686)

[Figure 28:ICES ACADEMIA Web Package diagram. 94](#_Toc147485687)

[Figure 29:Class Diagram Formalism 95](#_Toc147485688)

[Figure 30: ICES ACADEMY Class diagram 98](#_Toc147485689)

[Figure 31:Components of a Deployment Diagram 104](#_Toc147485690)

[Figure 32:ICES ACADEMY Deployment diagram 106](#_Toc147485691)

[Figure 33:Formalism of a Component Diagram 107](#_Toc147485692)

[Figure 34:Component diagram of our website 109](#_Toc147485693)

[Figure 35:view of DB testing 118](#_Toc147485694)

[Figure 36:Xampp Server Installation 123](#_Toc147485695)

[Figure 37:Apache setup wizard welcome page 124](#_Toc147485696)

[Figure 38:Apache component installation 125](#_Toc147485697)

[Figure 39:installation directory selection 125](#_Toc147485698)

[Figure 40:Installation Sponsor 126](#_Toc147485699)

[Figure 41:Ready to install 126](#_Toc147485700)

[Figure 42:installation progress 127](#_Toc147485701)

[Figure 43:installation end 127](#_Toc147485702)

[Figure 44:installation completed 128](#_Toc147485703)

[Figure 45:Home page view 129](#_Toc147485704)

[Figure 46:ADMIN PAGE VIEW 129](#_Toc147485705)

LIST OF TABLES

[Table 1: Hardware Resources found at ICES\_SARL 14](#_Toc147485896)

[Table 2: Software resources found at ICES\_SARL 16](#_Toc147485897)

[Table 3:Criticism of the existing system 25](#_Toc147485898)

[Table 4 Project's Plan 32](#_Toc147485899)

[Table 5 Software resources of the project 35](#_Toc147485900)

[Table 6 hardware resources of the project 36](#_Toc147485901)

[Table 7 Human Resources of the project 36](#_Toc147485902)

[Table 4:Comparism between MERISE and UML 49](#_Toc147485903)

[Table 5:COMPARATIVE STUDY BETWEEN 2TUP AND UP 51](#_Toc147485904)

[Table 6:Components of Use Case Diagram 55](#_Toc147485905)

[***Table 7:Components of Use Case Diagram (Continued)*** 56](#_Toc147485906)

[Table 8:System Actor's 57](#_Toc147485907)

[Table 9:Formalism of Use Case Specification (Textual Description) 62](#_Toc147485908)

[Table 10:SignUp Textual description 63](#_Toc147485909)

[Table 11:Login Textual Description 65](#_Toc147485910)

[Table 12: Communication diagram components 67](#_Toc147485911)

[Table 13:Sequence diagram components 70](#_Toc147485912)

[Table 14:Components of an Activity Diagram 75](#_Toc147485913)

[Table 15:State Machine Diagram Components 89](#_Toc147485914)

[Table 16:Components of a Package diagram 93](#_Toc147485915)

[Table 17:Components of a Class diagram 96](#_Toc147485916)

[Table 18: Deployment Diagram components 105](#_Toc147485917)

[Table 19:component of component diagram 108](#_Toc147485918)

[Table 20:Software resources used in project 110](#_Toc147485919)

[Table 21:Hardware Resources 111](#_Toc147485920)

[Table 22:Language used in project 111](#_Toc147485921)

[Table 23:Testing Methods 117](#_Toc147485922)

* 2TUP: Two track unified process
* ADMIN: Administrator
* AICS: African Institute of Computer Sciences
* CSS: Cascading style sheet
* DB: Database
* DBMS: Database management system
* HTML: Hypertext mark-up language
* Mr: Mister
* MVC: model view controller
* PHP: Hypertext pre-processor
* SQL: Structured query language
* UML: Unified modelling language
* UP: Unified process

GLOSSARY

Online certification training platforms are becoming more popular and accessible as a way of acquiring new skills and credentials. However, managing such platforms can be challenging, especially when dealing with large numbers of learners, courses, instructors, and certificates. This report presents a computerized management system that aims to simplify and automate the tasks involved in running an online certification training platform. The system consists of four modules: learner management, course management, instructor management, and certificate management. Each module provides various functions and features to facilitate the registration, enrolment, progress tracking, evaluation, feedback, and certification of learners and instructors.

ABSTRACT

Les plateformes de formation en ligne certifiantes sont de plus en plus populaires et accessibles pour acquérir de nouvelles compétences et qualifications. Cependant, la gestion de ces plateformes peut être difficile, surtout lorsqu’il s’agit de gérer un grand nombre d’apprenants, de cours, d’instructeurs et de certificats. Ce rapport présente un système de gestion informatisé qui vise à simplifier et à automatiser les tâches impliquées dans le fonctionnement d’une plateforme de formation en ligne certifiante. Le système se compose de quatre modules : gestion des apprenants, gestion des cours, gestion des instructeurs et gestion des certificats. Chaque module fournit diverses fonctions et caractéristiques pour faciliter l’inscription, l’inscription, le suivi des progrès, l’évaluation, la rétroaction et la certification des apprenants et des instructeurs.

RESUME

GENERAL INTRODUCTION

Online certification training is a form of education that allows learners to acquire skills and credentials in various fields and domains through online courses and assessments. Online certification training can offer many benefits, such as flexibility, convenience, affordability, accessibility, and personalization. However, online certification training also poses many challenges, such as quality assurance, security, reliability, scalability, and user satisfaction.

Computerized management of an online certification training platform is the process of using software systems and applications to facilitate and automate the various aspects and functions of online certification training. Computerized management can help to improve the efficiency, effectiveness, and quality of online certification training. It can also help to reduce the costs, risks, and errors associated with online certification training. To obtain these objectives, we used 6 main sections or phases which are;

**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 I

INSERTION PHASE

Preface

This Phase involves the discovery and integration into our working environment. This part presents our welcoming and integration into the enterprise and introduces us to the enterprise where our internship was carried out. It also considers the presentation of the administrative and functional organisation, its missions, geographical location, and company resources, a brief presentation of the project, and a conclusion.

Contents

INTRODUCTION

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

CONCLUSION

INTRODUCTION

At the early stage of an internship, interns are filled with excitement and anxiety trying to feel, connect and know well their environment and the people in it. This is the moment were interns get to know other interns, get in touch with other employees in or out of their department and so on. This experience is very unique to each intern, and we express ours in this part.

3. WELCOMING AND INTEGRATION INTO ICES SARL

The short period of time where a student gain knowledge and opportunity to acquire working experience with respect to a particular field in a particular enterprise is very crucial in the life of a student. ICES\_SARL been the enterprise, granted us the privilege to access the opportunity to experience a line of work for our individual career paths.

Upon arrival on the Monday,5 July 2023 at ICES SARL we were warmly welcomed by the General Manager of ICES\_SARL Mr. Joseph Kobe who introduced to us the Internship Coordinator (Overall director of the internship program) Mr. Bernard by which through him we were assigned an individual professional supervisor.

4. GENERAL PRESENTATION OF ICES SARL

4.1 History

INTERNATIONAL COMPUTER AND ELECTRONIC SYSTEM SARL abbreviated as ICES\_SARL is an IT Training Centre founded by Mr. Joseph Kobe in the year 2002. This training centre works in partnership with Microsoft and a JYGA () LTD Canadian Group representative.

4.2 Mission

During the past years, ICES\_SARL (INTERNATIONAL COMPUTER AND ELECTRONIC SYSTEM SARL) has had significant growth in personnel who share the same vision and have as mission

* PROMOTING MICROSOFT TRAINING TO BOTH THE YOUNGER AND OLDER GENERATION.
* MAXIMIZING BUSINESS POTENTIALS WITH TECHNOLOGY.

4.3 Activities

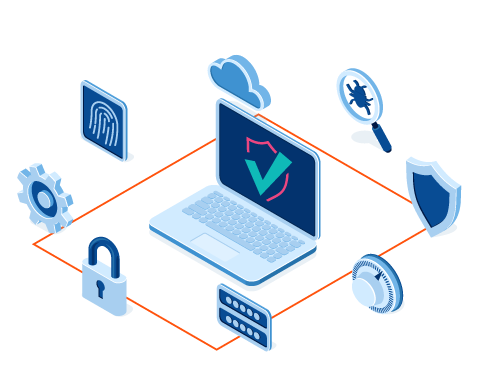
Due to their outstanding number of years in service, ICES \_SARL is made up of interconnected activities such as

* Consultation of Personnel
* Management of Different Internship programs
* Microsoft Certification training classes and Boot camps.
* Installation and sales of technological equipment such as cameras, routers, switches, and printers.
* Sales and Installation of Microsoft-related products such as Microsoft Azure, and Microsoft 365.

4.4 Services

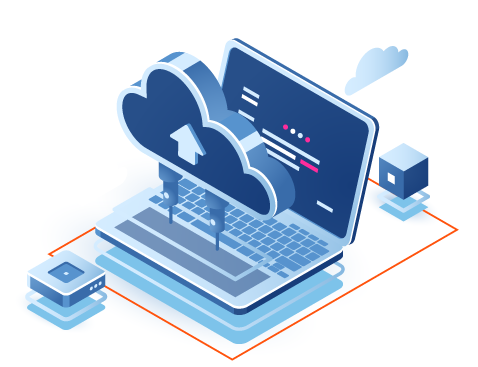
ICES SARL offers a Suit of Services and Products. Following is a noteworthy list of their services offered:

COMPANY INFASCTRUCTURE



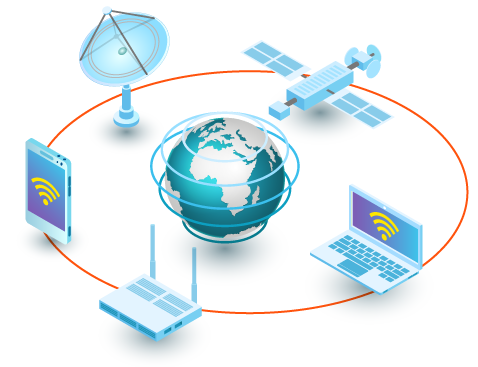
SECURITY

Protect the Integrity, ensure the security of company’s data and information from malicious sources.



VIRTUALIZATION AND CLOUD

Simplify Data Centre management, reduce operating costs ,eliminate downtime, increase IT productivity, agility and responsiveness with virtualization and cloud.



MANAGE NETWORK SOLUTIONS

ICES\_SARL also provides and implements fast, secure and reliable Managed network solutions for day to day businesses



SERVER SOLUTIONS

ICES\_SARL provides responsive and functional IT Design for business and companies worldwide

Figure : Company Infrastructure Services Provided by ICES\_SARL

4.5 Subscription Service

4.5.1 Microsoft Certification

This Subscription service is provided by Microsoft to students or individuals of different job role through ICES\_SARL who acts as a middleman between Microsoft and the different customers wishing to get certified. Some of these Certifications include:

* Microsoft Azure Certifications. Examples include Azure Fundamentals Certification (AZ-900).
* Microsoft 365 Certifications. Examples include Microsoft 365 Fundamentals (MS-900).
* Dynamics 365 Certifications. Examples include Dynamics 365 Fundamentals (MB-910).
* Power Platform Certifications. Examples include Microsoft Power Platform Fundamentals (PL-900).

4.6 JURIDICAL STATUS

International Computer and Electronic Systems also abbreviated ICES\_SARL is legally identified as a Limited Liability Company reason for their outstanding improvements in the past years since they are relatively easy to incorporate manage and operate.

4.7 GEOGRAPHICAL LOCATION

The LLC ICES\_SARL is located in Yaoundé precisely at BATA NLONGKAK. It can be easily located with a Red Marker below on the localization plan below.

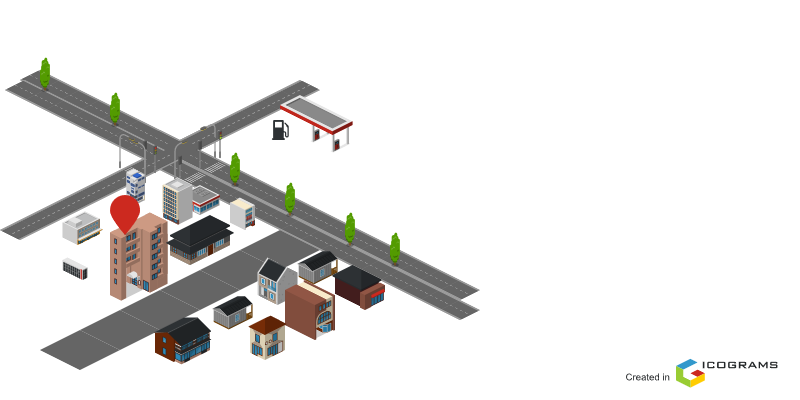


Figure : Geo Localisation of ICES\_SARL

5. PRESENTATION OF THE ADMINISTRATIVE AND FUNCTIONAL ORGANIZATION OF ICES\_SARL

Understanding how each department within an enterprise works and what they contribute to its success can be helpful in expanding both enterprise integration and knowledge. ICES\_SARL is organised as follows:

GENERAL MANAGEMENT

It is in charge of managing the day-to-day operations in the enterprise.

It has as mission:

* To draft the comprehensive work plan, summary, report documents.
* To comprehensively coordinate the daily administrative affairs management and the various departments of the enterprise.
* Managing the enterprise infrastructure.

Business Development

In short, Business Development department s responsible for customer interaction and service. They have as responsibility:

* Creating and implementing the enterprise’s sales strategy.
* Choosing the best tool and shaping sales process.

HR Department

They are in charge of managing employees according to the employee attendance policy

Network Administration

This department is in charge of managing, monitoring, maintaining, securing the enterprise’s network.

Some of its responsibilities include:

* providing alternative networks.
* diagnose and Troubleshoot network related problems.

Securing

This department protects enterprise assets and data artifact from potential security threats. They have as responsibility protecting and processing vital information.

Certification Training Department

It is in charge of managing the different certifications, trainees, instructors for different Microsoft certifications.

Organisation Chart Of ICES\_SARL

Below is the Organisational Chart of ICES\_SARL

Figure : Organization Chart of ICES\_SARL

6. RESOURCES

6.1 HARDWARE RESOURCES

Table : Hardware Resources found at ICES\_SARL

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| HARDWARE RESOURCES | | | | | |
| OUTPUT DEVICES | | | | | |
| Name | Manufacturer | | Reference | | General Description |
| HP LaserJet P2035 | HP | | Printer | | USB,600 x 600 dpi, 30ppm  Manual Duplex Printing, 300 Sheets Input capacity |
| HP Colour LaserJet Pro MFP M277n | HP | | Printer | | 600 x 600 dpi, 18ppm Colour Print,USB  150 Sheets Input Capacity |
| HP LE1901w | HP | | Monitor | | 19 inches, 1440 x 900 pixels  VGA |
| Dell 1907FPt | Dell | | Monitor | | 19" Fullscreen (1280 x 1024),  VGA, DVI, USB Hub,  Pivot, Swivel & Tilt Compatibility |
| DESKTOP | | | | | |
| Dell OptiPlex 780 | | Dell | Computer Device | | Intel® Core™2 Duo,  2 GB RAM  HDD 160GB |
| NETWORK DEVICE | | | | | |
| Huawei ETS2258 | Huawei | | Fixed Wireless Terminal | / | |
| Huawei B310 | Huawei | | Router | LTE Cat 4 DL 150/UL 50 Mbps,  Wi-Fi IEEE 802.11b/g/n with a range of 200 meters | |
| Microtik RB2011Ui | microtik | | Router | 128MB RAM, 128MB NAND,  300 Mbit/s | |

SOFTWARE RESOURCES

Table : Software resources found at ICES\_SARL

|  |  |  |
| --- | --- | --- |
| SOFTWARE RESOURCES | | |
| Operating Systems | Software | Antivirus |
| Windows 8 and above | Microsoft 365,  Microsoft Teams, Any Desk, WampServer,  Microsoft Office, chrome, Microsoft Edge, Bing, SharePoint, Android Studio,  Visual Studio, Microsoft Power Platform apps | Windows Security Avast, Norton, |
| Windows Server | Any Desk, Wireshark, Nmap | Microsoft Defender, Surf shark,  Nord Threat Protection,  Total AV |

7. BRIEF PRESENTATION OF PROJECT

As many people become more aware of the advantages and benefits certification adds to their professional and educational, many different training centres (certification training centre's) tend to grow in size as a result faces some numerous problems like increase in cost due to fact that all documents had to be photocopied thereby increasing expenses, classroom became small since number of students steeply increased. Some training centres thought as idea of doing shifts per day. Despite all these, training centre's still face some challenges unavoidable but manageable. Therefore, there is a requirement of a platform to manage, automatize and computerise the certification cycle. Hence our theme “**COMPUTERISED MANAGEMENT OF AN ONLINE CERTIFICATION TRAINING PLATFORM**". With our solution subscribers(students) can remotely attend a class, instructors and share e-documents and test without having to spend to photocopy. Also, student to teacher interaction is enhanced with the presence of live chats, we can also track and record student performance and finally notify students of their various certification due dates.

CONCLUSION

In general, insertion plays a vital role in the internship life cycle of a student as it possesses the student with a spirit of collaboration. To us, it was fabulous, enthusiastic, and fluid with countless workers including our professional supervisor. Next, we have the Existing System.

PART II

EXISTING SYSTEM

Preface

This document provides a complete description of our project, covering what it should and should not do, how it should look, what it can and cannot handle, and how it will store data. This document serves as a guide for the development process.

Contents

INTRODUCTION

1. Presentation of the project theme
2. Study of the existing system
3. Criticism of the existing system
4. Problematics
5. Proposed solutions

CONCLUSION

INTRODUCTION

We initiate our exploration by delving deep into the existing system. This serves a dual purpose: recognizing its merits and equally importantly, identifying areas where it falls short. By casting a critical eye, we don’t just identify problems; we open doors to potential improvements and innovative solutions. This section will articulate challenges, discrepancies, and the gaps we aim to bridge.

1. Brief Presentation of the project theme

As the demand for certification grows among professionals and students who recognize its value, many training centres face various challenges in delivering quality education. Some of these challenges include high costs of photocopying documents, limited classroom space, and scheduling conflicts. To overcome these difficulties, there is a need for a platform that can automate and computerize the certification process. This is the aim of our project: “**COMPUTERIZED MANAGEMENT OF AN ONLINE CERTIFICATION TRAINING PLATFORM case study ICES SARL**”. With our solution, subscribers (students) can access online classes, instructors, e-documents, and tests without spending money on photocopying. They can also interact with teachers through live chats, track and record their performance, and receive notifications about their certification deadlines.

1. STUDY OF THE EXISTING SYSTEM
2. CRITICISM OF THE EXISTING SYSTEM

Table :Criticism of the existing system

|  |  |  |
| --- | --- | --- |
| PROBLEM | CONSEQUENCES | PROPOSED SOLUTION |
| USERS | | |
| Ineffective management of courses | This sometimes leads slow progression of students. | We propose a form to allow teachers access a platform to be able to manage the different courses |

### 

1. PROBLEMATICS

During the COVID-19 ERA, the rate at which e-learning was used, was very high due to the fact that students couldn’t really get close to each other. On our arrival to ICES we came across a similar situation and we taught of creating a platform which can help students interact with each other remotely.

And there comes the purtubing question; **How can we create a certification management platform to ease the collaboration and interaction of students and instructors.**

1. PROPOSED SOLUTION

Based on the problematic and the study of the current system in Cameroon, we decided to implement a software solution that will aim to:

* Help teachers manage their courses
* Help students easiy access the courses
* Help teachers manage their different course assessment

CONCLUSION

In this phase we were to present the existing system and see their down fall and proposed a solution that helps to bridge the gap in the society that is faced by Cameroonian population and through these we are now going forth to the specification book.

PART III

SPECIFICATION BOOK

**PREAMBLE**

This document outlines the comprehensive specifications for our project, detailing functional and non-functional requirements, UI design, system limitation, and database structures serving as the development guide.

**CONTENT**

INTRODUCTION

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

CONCLUSION

## INTRODUCTION

The specification book sets the foundation for understanding the software project at hand. It provides an overarching summary of the software’s purpose, the scope of its functionality and the specific objectives it aims to achieve. It also lays out the intended audience of the software, any assumptions or prerequisites for its use, and the strategic approach to its development. Following this, the specification book delves deeper into the intricate details of the software, covering aspects like functional and non-functional requirements, system limitations and constraints, preliminary user interface design, and data structures of database design. In this document, it will serve as a pivotal roadmap guiding the software development process. We will also include project details such as the team involved, constraints, the budget, deadlines, and deliverables.

### CONTEXT AND JUSTIFICATION OF STUDIES

The rapid development of information and communication technologies (ICT) has created new opportunities for learning and professional development. Online learning platforms are one of the most popular and effective ways to deliver high-quality education and training to learners across the world. Online learning platforms also provide learners with flexibility, convenience, and affordability, as they can access the courses anytime, anywhere, and at their own pace.

However, online learning platforms also face some challenges and limitations, such as:

* Managing a large number of courses, instructors, learners, and certificates in an efficient and secure way.
* Ensuring the quality and validity of the courses and certificates offered by the platform.
* Providing effective feedback and support to the learners and instructors.
* Enhancing the engagement and retention of the learners and instructors.
* Evaluating the impact and outcomes of the online learning programs.

To address these challenges and limitations, the project COMPUTERIZED MANAGEMENT OF AN ONLINE CERTIFICATION TRAINING PLATFORM aims to design and develop a software system that can automate and optimize the processes and functions of an online learning platform. The project will use the latest technologies and best practices to create a user-friendly, reliable, and scalable system that can handle the following tasks:

* Course creation and management: The system will allow instructors to create and upload courses, assign quizzes and assignments, set deadlines and grading criteria, and monitor the progress and performance of the learners.
* Learner registration and management: The system will allow learners to register for courses, access course materials, submit quizzes and assignments, receive feedback and grades, and track their learning outcomes.
* Certificate issuance and verification: The system will generate certificates for learners who successfully complete the courses, based on predefined criteria. The system will also provide a mechanism for verifying the authenticity and validity of the certificates issued by the platform.
* Feedback and support: The system will provide various channels for communication and interaction between learners, instructors, and platform administrators. The system will also provide automated feedback and guidance to the learners based on their performance and needs.
* Analytics and reporting: The system will collect and analyse data from various sources, such as course content, learner behaviour, quiz results, assignment submissions, feedback ratings, etc. The system will generate reports and dashboards that can help evaluate the effectiveness and impact of the online learning programs.

### OBJECTIVES OF THE PROJECT

#### General objectives

To design and implement a digital platform that aids in the efficient recovery and management of lost ID cards in Cameroon.

#### Specific Objectives

In order to achieve the main objective, there are specific objectives to be achieved.

* **User Experience**: Create a user-friendly interface where individuals can sign up and login and easily.
* **Notifications**: Establish an effective notification system that admin can use to alerts the users
* **Safety and Security**: Ensure that the platform protects users ‘sensitive information, maintaining confidentiality while also being accessible for recovery purposes.
* Name of the project: **ICES ACADEMY**
* Project target: ICES SARL
* Technical specification : WEB PLATFORM mainly.

EXPRESSION OF NEEDS

In this section we are going to outline the specific needs or requirements that the software project is intended to meet. It is a bridge between the problem and the solution detailing the ‘what’ and not the ‘how’. And this need is going to be divided into functional and non-functional needs.

Functional needs

Here, we are talking about **what** the system is supposed to do for the user of the application, and these features are.

* User should signup and sign in (we should be able to identify the user)
* User can fill Automatically create and modify course assessment
* User can take a course assessment in the platform
* User can get notification in case a certification is outdated

NON-FUNCTIONAL NEEDS

The non-functional need we are referring to here, is the ‘quality attributes’ of the software, that is, how the software should behave in order to increase user expectations and satisfaction and just below we are going to see the non-functional need that our application needs to have.

* **Security:** User personal data should be securely stored and transmitted in order to respect user privacy.
* **Performance:** The application should be able to load quickly and operate smoothly to ensure maximum user satisfaction
* **Usability:** it must be easy to use and navigate in order to avoid making the app seems complicate to use by the user
* **Reliability:** it must be reliable in order to meet up market expectation and to promote the needs of the application

### PLANNING OF THE PROJECT

Table Project's Plan

|  |  |  |  |
| --- | --- | --- | --- |
| Week | period | task | method |
| INSERTION PHASE | | | |
| Weeks 1-2 | **Morning**  (08h30 – 12h) | Documenting on the structure of  the organization | theoretical |
| **Afternoon**  (13h-17h) | 1. General presentation of Hadron    * History and Evolution of ICES SARL    * Missions and activities    * Key Realization    * Geographical location 2. Administrative and Functional Organization of ICES SARL 3. Hardware and Software Resources of ICES SARL 4. Brief Presentation of the project theme   Conclusion (13h-17h) | practical |
| TECHNICAL PHASE | | | |
| Weeks 3 Specification | Morning  (08h30 – 12h) | Understanding the themes   * Ask frequent questions to understand the basic needs. * Understand the basic objectives of the application with the stakeholder | practical |
| Afternoon (13h-17h) | Requirements   * Gathering and defining the requirements * Documenting the specifications accurately | Theoretical/practical |
| Weeks 4-7  Analysis phase | Morning  (08h30 – 12h) | Documenting on the method of  analysis (UML), diagrams and | Theoretical |
| Afternoon (13h-17h) | Implementing on the detailed  structure of the application using UML diagrams | Practical |
| Weeks 8-9 Design | Morning  (08h30 – 12h) | Documenting the application architecture and  component/modules and UI | Theoretical |
| Afternoon  (13h-17h) | Designing the components and UI | Practical |
| Week 10-12 deploying and testing the application | Morning  (08h30 – 12h) | * Writing the source code * Setting up the   development environment | Practical |
| Afternoon (13h-17h) | * Preparing the application for deployment * Testing the application for deployment | Practical |

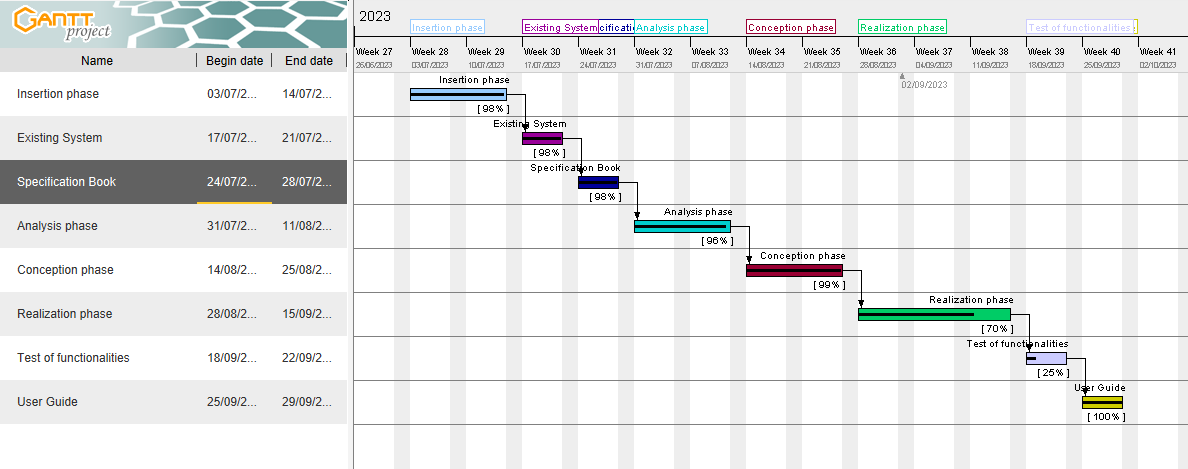


Figure 4: Gantt projectt

### ESTIMATION OF THE PROJECT

#### SOFTWARE RESOURCES

Table 5 Software resources of the project

|  |  |  |  |
| --- | --- | --- | --- |
| SOFTWARE | USAGE | QUANTITY | PRICE(FCFA) |
| Microsoft office 2016 | Used for the creation of the report and power point | 1 | 287,500 |
| Visual studio | Open and free software for the coding part of the project | 1 | Free |
| PowerAMC | For doing the design of the diagrams to represent the structure and the interaction between the different actors | 1 | 85000 |
| Microsoft edge | For doing the research on the internet and testing of the application | 1 | Free |
| Windows 11 (64 bits) HOME | Operating system | 1 |  |
| Gannt Project | For building a gannt chart for the planning of the work | 1 | free |
| Mysql | For creating database | 1 | Free |
| TOTAL 1 |  |  | 372,500 FCFA |

#### HARDWARE RESOURCES

Table 6 hardware resources of the project

|  |  |  |  |
| --- | --- | --- | --- |
| MATERIAL | USAGE | QUANTITY | PRICE (FCFA) |
| LAPTOP (HP CORE i5 11gen, SSD) | Used to build the system, type the report, create the power point, make research, etc. | 1 | 700,000 |
| USB flash disk 32gb | Used for storage and to print the report | 1 | 16,000 |
| Modem camtel | Used for internet connection | 1 | 65,000 |
| Mobile phone (iPhone XR) | Used for carrying research and storage | 1 |  |
| Printer HP Officejet 4630 | Used to print the scientific document | 1 | 86,520 |
| Total 2 |  |  | 867,520 FCFA |

#### HUMAN RESOURCES

Table 7 Human Resources of the project

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Role | Pay (FCFA)/day | Number | Duration(days) | Price (FCFA) |
| Project manager | 250000 | 1 | 60 | 15,000,000 |
| Analyst | 300000 | 1 | 30 | 9,000,000 |
| Designer | 500000 | 1 | 14 | 7,000,000 |
| Developer | 485000 | 1 | 35 | 16,975,000 |
| TOTAL 3 |  |  |  | 47,975,000 FCFA |

|  |  |  |  |
| --- | --- | --- | --- |
| TOTAL 1 | TOTAL 2 | TOTAL 3 | TOTAL (FCFA) |
| 372500 | 867520 | 47975000 | 49,215,020 FCFA |

Total in letter:

Forty-nine million two hundred and fifteen thousand and twenty Francs CFA

### PROJECT CONSTRAINTS

#### CRITERIA OF ACCEPTABILITY

The delivered project is judge acceptable if it respects all the criteria, that is the functionality that was presented.

#### TIME CONSTRAINTS

The project will be realized within 3 months that is the time allocated by the school for the internship and realizing of the project that is from July to September

### DELIVERABLES

At the end of this project, we are expected to submit the following:

A complete report containing:

* A User’s guide of the application
* A CD-ROM containing:
* The applications’ source code.
* Database (exported sql file)

## CONCLUSION

This section was a great step in the accomplishment of the project, and we were expected to clarify the different stakeholders the effectiveness and the use of the application in the market and its necessities by giving a detail structure of how the application will behave and help Cameroonians and give the estimation of the project cost for its realization and date of accomplishment of the project.

PART IV

ANALYSIS PHASE

Preface

The Analysis phase includes the collection of the customer requirements to present a solution

fine-tuned to their needs. The main purpose of this phase is to capture the user’s needs and define processes and prominent methodologies used. This phase includes an Overview of prominent methodologies followed by Justification of methodology choice and finally modelling of the system.

Contents

1. INTRODUCTION

2. PRESENTATION OF ANALYSIS METHOD

3. JUSTIFICATION OF ANALYSIS METHOD

4. SYSTEM MODELING

5. CONCLUSION

INTRODUCTION

Outstanding projects start with an outstanding breadth of view. Like any other multifaceted system, software development calls for dive-deep planning, groundwork, and management, especially, if you have to juggle several processes. Therefore, any good developer worth his salt should be well-versed in the methodologies and choose the one according to the project context and the business requirements.

2. PRESENTATION OF ANALYSIS METHODS

2.1 OVERVIEW OF PROMINENT METHODOLOGIES

I. UML

UML stands for **U**nified **M**odelling **L**anguage. It’s a rich language to model software solutions, application structures, system behaviour and business processes. It consists of 14 diagrams.

II. MERISE

Merise is an integrated analysis, design and project management method, which is its main strength. It was first early introduced in the 1980s.

III. Agile

The agile methodology created in 2001 is a project management approach that prioritizes cross-functional collaboration and continuous improvement. It divides projects. into smaller phases and guides teams through cycles of planning, execution, and evaluation.

The agile method concept is the result of the Agile Manifesto in which we find four

fundamental values, and twelve principles defined as good practice rules.

IV. SCRUM

Introduced in 1995 is an empirical process, where decisions are based on observation, experience, and experimentation. Much like a rugby team (where it gets its name) training for the big game, scrum encourages teams to learn through experiences, self-organize while working on a problem, and reflect on their wins and losses to continuously improve. It has three pillars: **transparency**, **inspection,** and **adaptation**.

V. RAD (RAPID APPLICATION DEVELOPMENT)

**Rapid application development (RAD) created in the 1980s is an agile software development approach** that focuses more on **ongoing software projects** and **user feedback** and less on following a strict plan. It includes a 4-phase development cycle: requirements, design, construction, and application generation.

VI. XP (eXtreme Programming)

Extreme programming is an Agile project management methodology created in the late 1990s that targets speed and simplicity with short development cycles and less documentation. The process structure is determined by five guiding values, five rules, and 12 XP practices.

VII. UP (Unified Process)

It is an iterative and incremental software development methodology controlled by the UML use cases. The Engineering Requirements present in the Unified Process are mostly technology-centric and have only recently focused on user-centered design. UP is organised into four major phases: Inception, Elaboration, Construction, and Transition.

2.2 INTRODUCING UML LANGUAGE

UML, short for **Unified Modelling Language**, is a standardized modelling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modelling and other non-software systems. This language includes a set of graphical notations(diagrams) techniques to create visual models of software systems.

UML is a notation that resulted from the unification of OMT from **Object Modelling Technique OMT** - James Rumbaugh 1991 (Best for analysis and data-intensive information systems), **BOOCH**-Grady Booch 1994(Excellent for design and implementation), OOSE Object-Oriented Software Engineering - Ivar Jacobson 1992(Featured a model known as Use Cases. Powerful technique for understanding the behaviour of an entire system.).

In 1994, Jim Rumbaugh, the creator of OMT joined Grady Booch at Rational Corp. The partnership aimed to merge their ideas into a single, unified method (the working title for the method was indeed the "**Unified Method**").

By 1995, the creator of OOSE, Ivar Jacobson, had also joined Rational, and his ideas (particularly the concept of "Use Cases") were fed into the new Unified Method - now called the Unified Modelling Language1. Due to the dissatisfaction and limitations of the previous version developers set up a working group to build and improve the previous version to other versions such as UML 1.5 (Consisting of 9 diagrams). Today the latest version of the specification validated by the OMG in 2017 is UML 2.5.1 which includes 14 diagrams categorized into **Structural Diagrams** and **Behavioural Diagrams**

a. Structural Diagrams.

Structural diagrams show the building blocks of our system; they portray features which don’t change with time. They include;

* Class diagram.
* Object diagram.
* Composite structure diagram.
* Deployment diagram.
* Component diagram.
* Package diagram, Profile diagram.

b. Behavioural Diagrams.

Behavioural diagrams show how our system responds to requests or otherwise evolves over time. They include;

* Use Case diagram.
* Activity diagram.
* State machine diagram/Protocol state machine diagram.

In this category of behavioural diagrams, we also have diagrams called **Interactive diagrams.**

Interaction Diagrams.

Interaction diagrams depict the exchange of messages within a collaboration (a group of cooperating objects) en route to accomplishing its goal. They include;

* Sequence diagram.
* Communication diagram.
* Overview diagram.
* Timing diagram.

c. UML Diagrams Overview

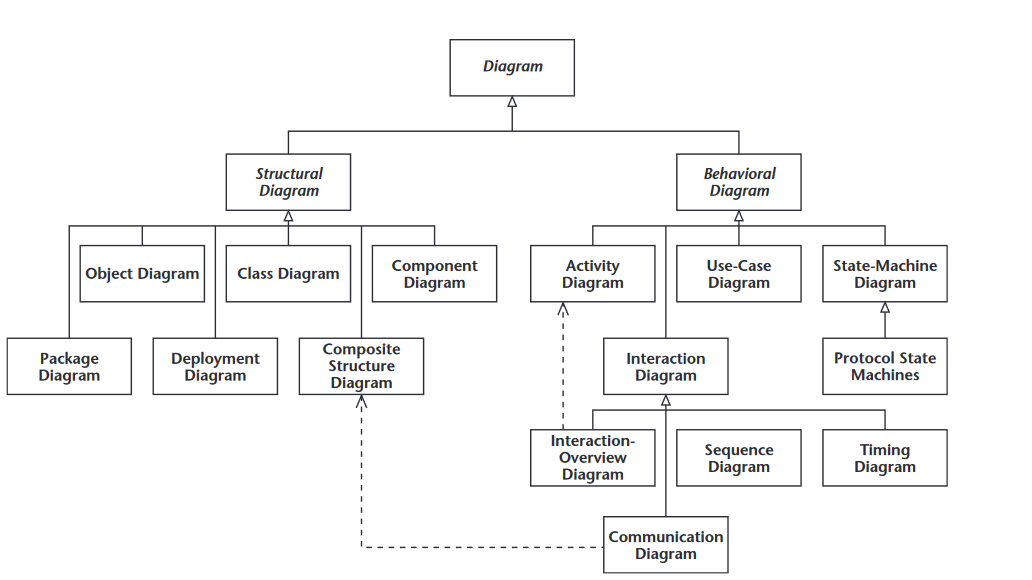


Figure :UML Diagrams Overview (Source: UML 2 For Dummies,2003 p.19)

It is important point to note that UML doesn’t recommend any approach therefore it is not a method but a modelling language. As such to give it an approach we need to associate UML to a Unified Process (UP) in other to give our conception a methodology to follow.

d. COMPARATIVE STUDY BETWEEN MERISE AND UML

Table :Comparism between MERISE and UML

|  |  |
| --- | --- |
| MERISE | UML |
| MERISE stands for Méthode d'Étude et de  Réalisation Informatique pour les Systèmes  d'Entreprises. | UML stands for Unified Modelling Language. |
| MERISE is a method for designing processing- and data-oriented systems. | UML is a modelling language, not a method. It needs a method that uses the UML language to design the system. (e.g. UP, RUP, 2TUP and possibly agile methods). |
| Less widely used | Widely used |
| Less complex and little or no time-consuming. | Complex and time-consuming due to the many diagrams it has. |
| Design for Organizational Information Systems. | Designed for Object-Oriented-based Information Systems. |

2.2 SOFTWARE DEVELOPMENT PROCESS

software development process is an iterative logical process that aims to create software which meets unique business or personal objectives, and goals. It permits us to obtain software systems or upgrades of previous or existing ones.

a. The Unified Process (UP)

The Unified Process is an iterative, incremental, architecture-centric, and use-case-driven approach to software development built on UML. UP is organised into four major phases:

Inception: It has as objective preparation of project basis, business case, establishing project scope and setting boundaries.

Elaboration: The majority of the systems requirements (e.g. in the form of a use case) are captured in this phase and a plan of risk management is made to reduce or eliminate their impact on the final schedule and product.

Construction: The final product is constructed, including the remaining lower-risk and easier elements of the system, again in an iterative fashion.

Deployment: During this phase, the new system is delivered to its end users. This phase includes data migration from legacy systems and user training.

b. TWO TRACKS UNIFIED PROCESS (2TUP)

2TUP is a software development process that implements the unified process (i.e. iterative, incremental, based on UML). It proposes a development cycle which separates the technical aspects from the functional aspects. It begins with a preliminary study which essentially consists of identifying the actors and the system exchange. Then to produce the specifications and to model the context. The software development process is structured around three branches:

* Functional branch (the left-hand branch of the Y).
* Technical branch (the right-hand branch of the Y).
* Realization branch (the middle branch).

Functional branch (the left-hand branch of the Y).

It captures the functional needs and produces a model of needs focused on the business of the users. Also, the analysis consists of precisely studying the functional specification in order to obtain an idea of what the system will achieve in terms of business.

Technical branch (the right-hand branch of the Y)

The technical branch lists the technical needs and proposes a generic design validated by a prototype. It lists all the constraints to be met in order to build the system. It then defines the components needed to build the technical architecture. This branch is independent of the functional aspects.

Realization Branch (the middle branch).

This branch consists of bringing the two branches together, allowing application design to be carried out and finally the delivery of a solution adapted to the needs. This branch includes:

**The preliminary design stage:** This step produces the system design model. This model organizes the system into components, delivering technical and functional

functional services. This induces the regrouping of the information from the technical and functional branches.

**Detailed design:** Allows you to study how to make each component. The result is a ready-made image of the complete system.

**Coding stage**: Permits us to effectively produce the components and progressively tests the code units produced as they are completed.

**The acceptance(recipe) stage:** It finally consists in validating the functions of the developed system.

i. COMPARATIVE STUDY BETWEEN 2TUP AND UP

|  |  |
| --- | --- |
| 2TUP | UP |
| 2TUP is a software development process that implements the unified process (i.e. iterative, incremental, based on UML). | The Unified Process is a generic name for a family of process models that meet a number of criteria, such as being iterative and incremental, driven by use cases, and focusing on addressing risks early. It defines four project phases: Inception, Elaboration, Construction, and Transition. |

Table :COMPARATIVE STUDY BETWEEN 2TUP AND UP

Overview of 2TUP

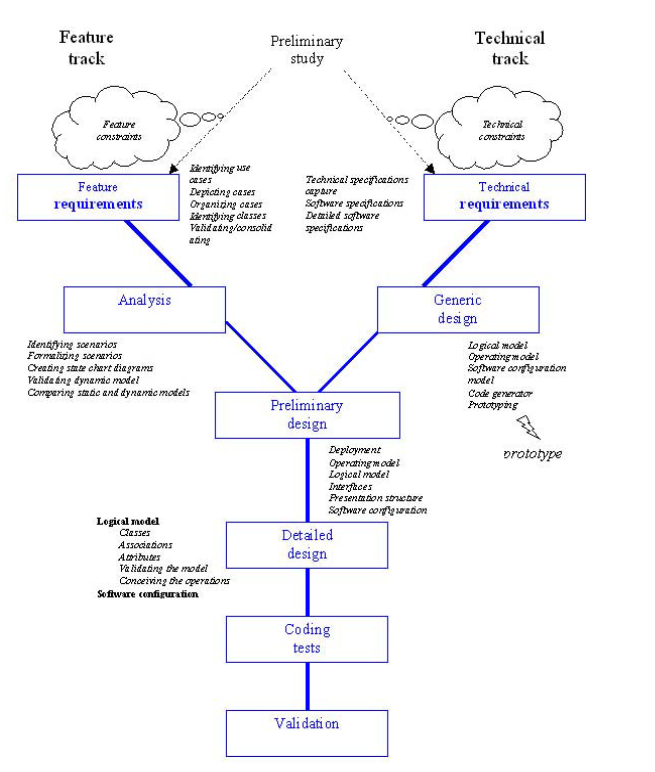


Figure :Overview of 2TUP(Source: <https://cedric.cnam.fr/fichiers/RC967.pdf> p.3)

3. JUSTIFICATION OF ANALYSIS METHOD

UML satisfies an important need in software and system development. Listed below are the reasons for our choice;

* UML is Designed for object-oriented Information systems.
* UML is based on the concept of information hiding and abstraction. Describe the ability of modellers (or their tools) to suppress or hide known information from a diagram to accomplish a goal (such as simplicity or repurposing).
* UML modelling also supports multiple views of the same system.

Just as you can have a political map, a relief map, a road map, and a utility map of the same area to use for different purposes — or different types of architectural diagrams and blueprints to emphasize different aspects of what you’re building — you can have many different types of UML diagrams, each of which is a different view that shows different aspects of your system.

* UML also allows you to construct a diagram for a specialized view by limiting the diagram elements to a particular purpose at a particular time.
* 2TUP is a software development process built on the UML modelling language.
* 2TUP is an incremental process.
* 2TUP is an iterative process. The degrees of abstraction are increasingly precise at each iteration.
* 2TUP is user-oriented because built on their expectations (i.e. permits the development of software that responds to users' needs).

4. SYSTEM MODELLING

4.1 FUNCTIONAL BRANCH

This essentially consists of identifying the actors who will interact with the system to be built, the messages that the actors and system exchange producing specifications and modelling the context.

1. Use Case Diagram

a.) Definition

A Use case diagram captures a system's high-level functions(requirements), dynamic nature and scope.

Use Case diagram serve for the following purposes;

* Specify the context or scope of a system.
* Capture the requirements of a system.

b.) Formalism

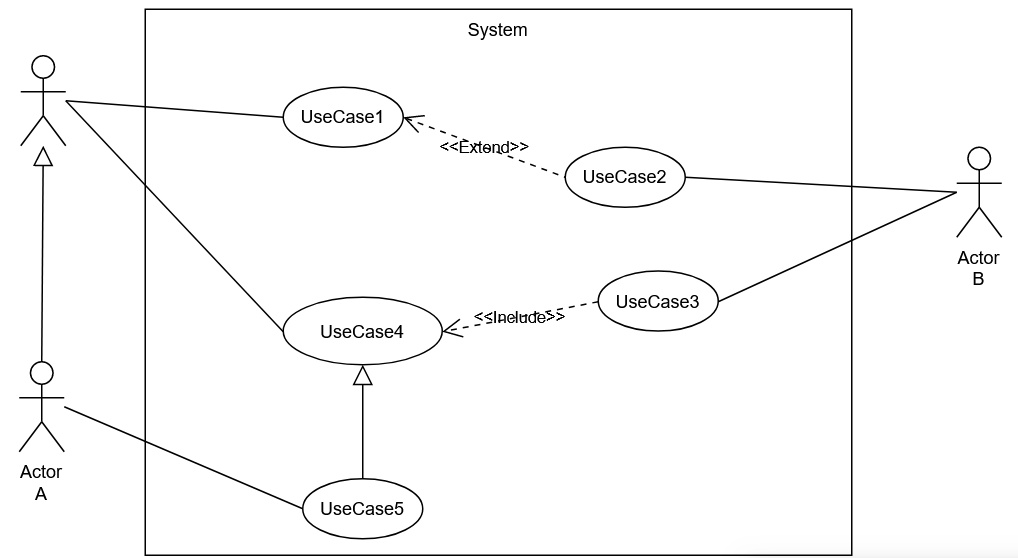
 Figure :UseCase Diagram Formalism

Table :Components of Use Case Diagram

|  |  |  |
| --- | --- | --- |
| ELEMENT | NOTATION | DESCRIPTION |
| Actor |  | Any Entity external to a system interacts with the system, maybe a human user or another system, and has goals and responsibilities to satisfy  in interacting with the system. |
| Use Case |  | A use case corresponds to a system's objective motivated by the actors' needs. |
| Association |  | It expresses the interaction between an actor and a use case. |
| Include |  | An Include Association from one use case (called the **Base use case**) to another use case (called the **inclusion use case**) indicates that the base use case will include or call the inclusion use case. |
| Extend |  | An Extend Association indicates that the extension use case will extend into)and augment the Base use case. |

***Table 11:Components of Use Case Diagram (Continued)***

|  |  |  |
| --- | --- | --- |
| ELEMENT | NOTATION | DESCRIPTION |
| Generalisation |  | An actor generalization from a specialized, actor to a generalized, actor indicates that instances of the more specific actor may be substituted for instances of the more general actor. |
| System |  | It identifies what is part of the system and the actors interacting with it. |

C. System’s Actors

Use case diagrams are considered for high-level requirement analysis of a system. So when the requirements of our system (ICES ACADEMY) are analysed the functionalities as well as actors, were captured. Below is a table list of our System Actors.

Table :System Actor's

|  |  |
| --- | --- |
| Actor | Role |
| Visitor | Visit the platform |
| Student | Verified individuals who interact with our platform in order to benefit from the service suit. |
| Instructor | These are verified Professionals with the responsibility to collaborate with or Consell students in the platform. |
| Administrator (Admin) | Their main role is to manage and control the system's functioning. |
| Messenger API | Provide our system with online Student-to-teacher service. |
| Microsoft Teams | Provides our system with remote lectures online. |

d.) General Use Case Diagram

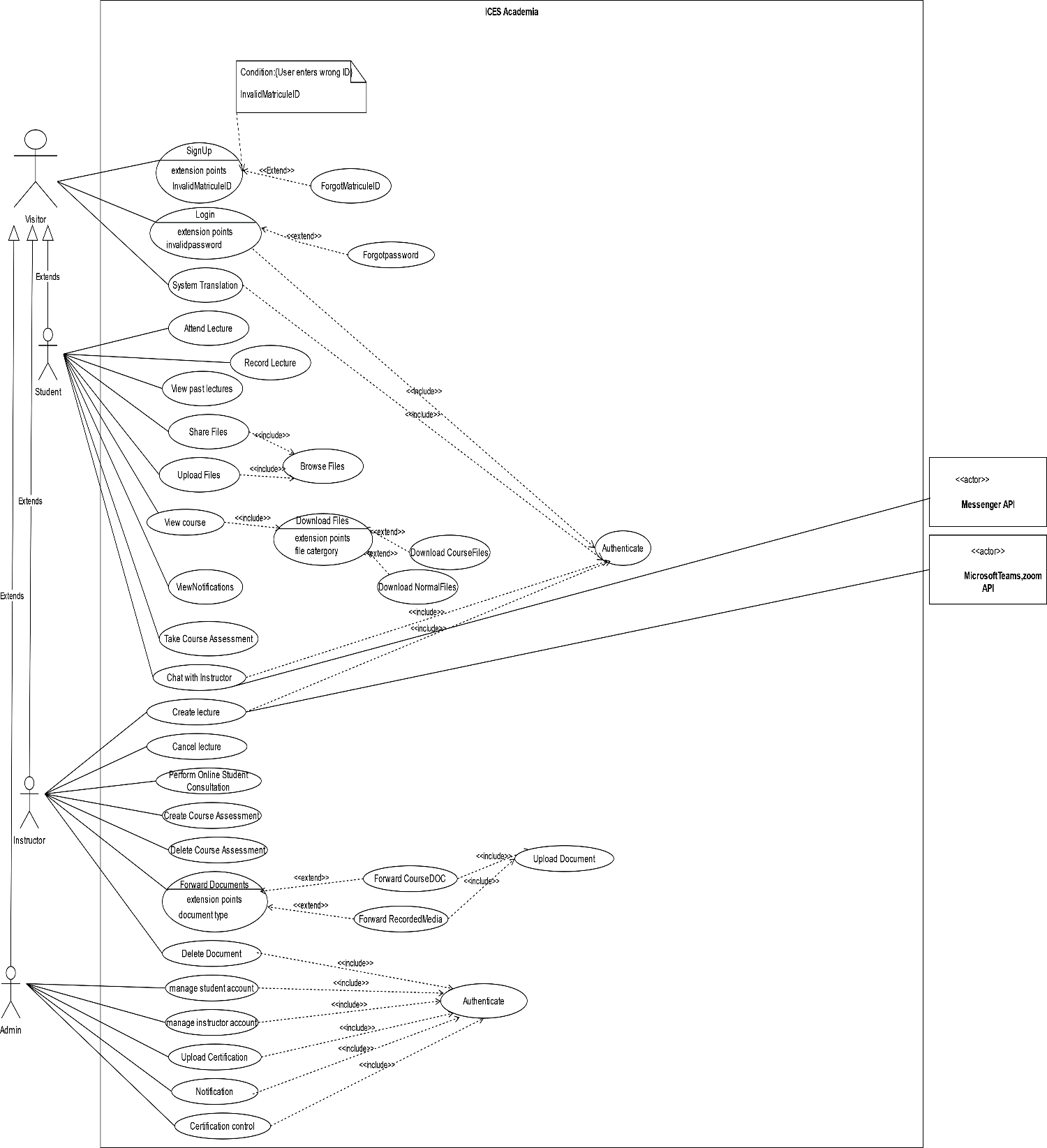


Figure :General Use Case Diagram

e.) Student Overview Use Case Diagram

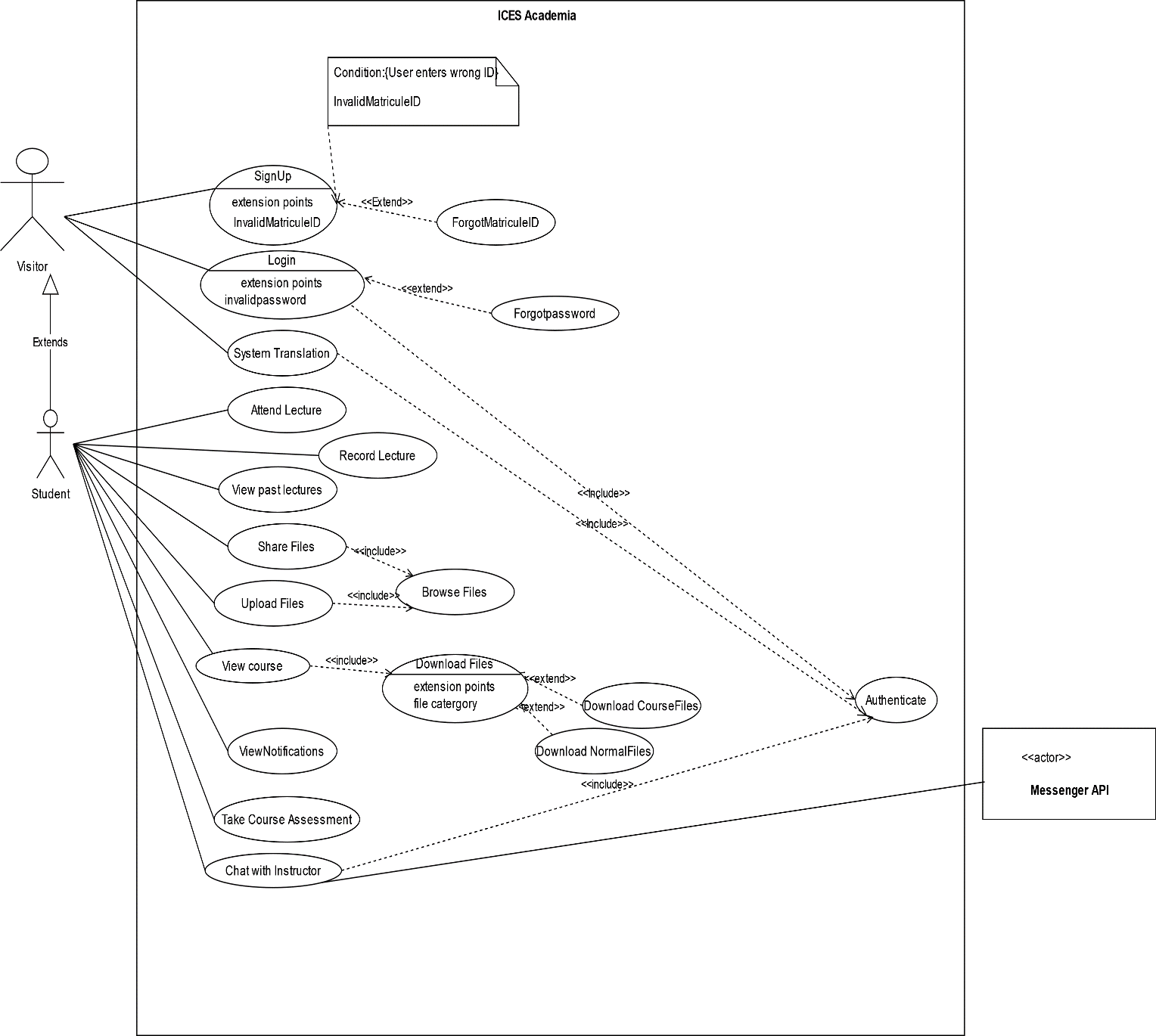


Figure :Student Overview Use Case diagram

f.) Instructor Overview Use Case Diagram

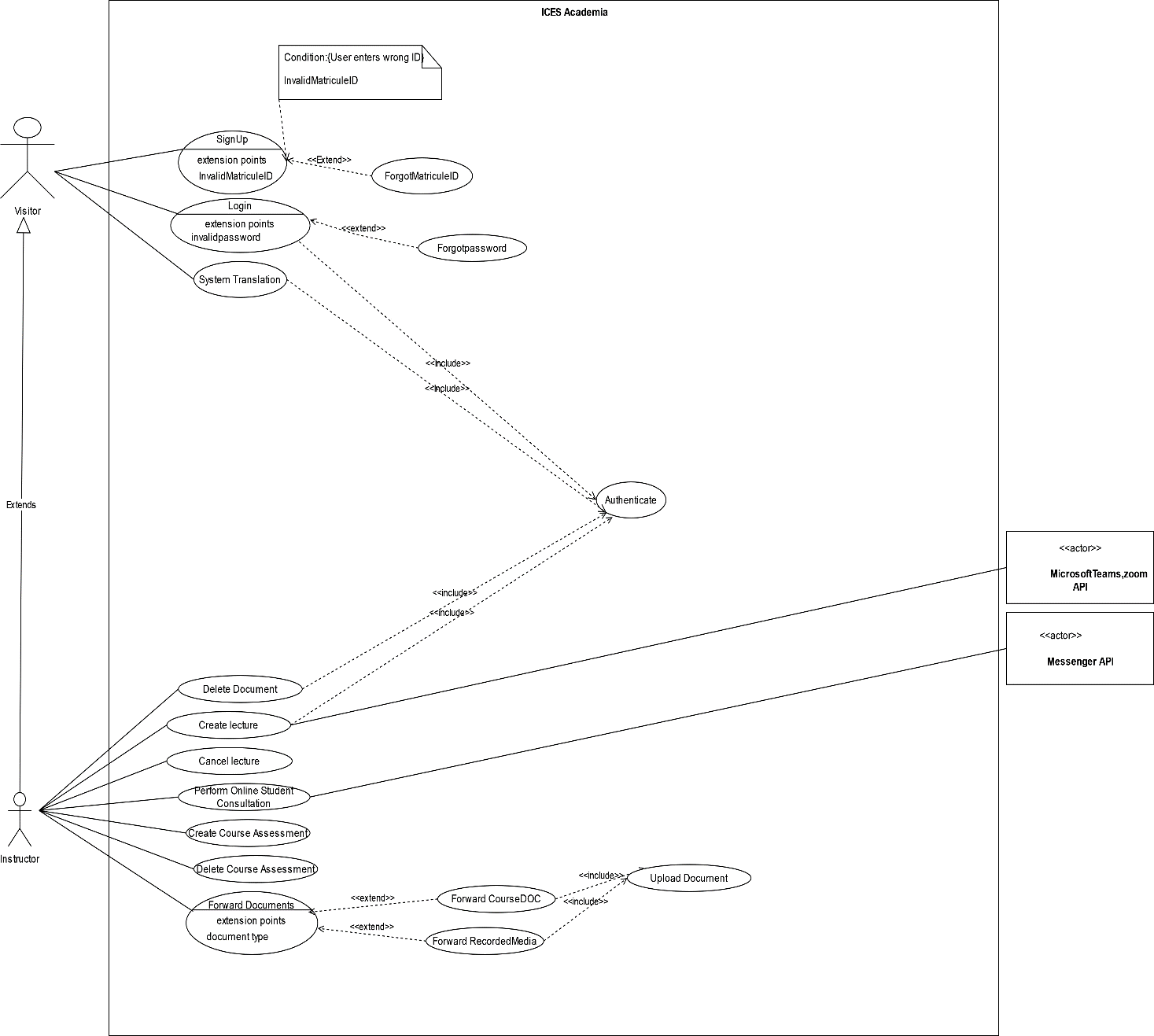


Figure :Instructor Overview Use Case Diagram

g.) Admin Overview Use Case Diagram.

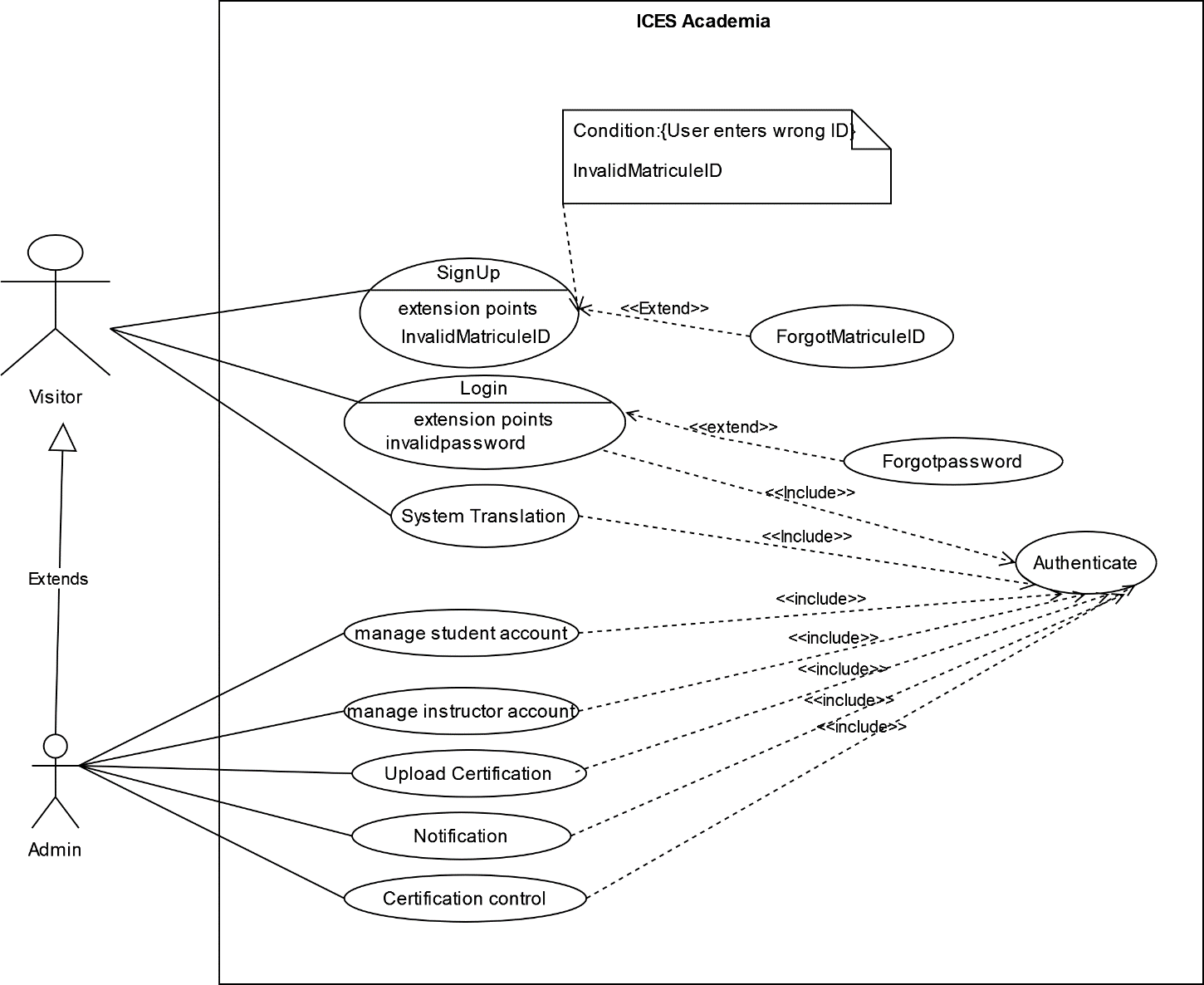


Figure :Admin Overview Use Case Diagram

h.) Use Case Specification (Textual description)

Use Case can be considered as a behaviour that a system offers to the actors to help meet the actors’ goals. So, we have to supply information on how the use case works — and put that information somewhere in a Textual description (Specification). This set of needed details placed inside a use case is sometimes called the **Use Case Textual Description.**

Formalism

Table :Formalism of Use Case Specification (Textual Description)

|  |  |  |  |
| --- | --- | --- | --- |
| Use-Case name: | | | |
| Description | It explains the goals, plot, and theme of the use case. | | |
| Actors | Description of the main and the secondary actors. | | |
| Date | Creation and update date. | | |
| Stakeholder | Names of creators. | | |
| Version | The version number. | | |
| Precondition | Specifies the state of the world that must be held before the course can be triggered. | | |
| Trigger | Any event that causes the use case to be initiated. | | |
| Main Course Scenario | | | |
| Primary Actor | | System | Secondary Actor |
| Alternative Scenario | | | |
| Primary Actor | | System | Secondary Actor |
| Post Condition of Success | Describes the state of the system after the end of each scenario. | | |
| Post Condition of Failure | Describes the state of the system after the end of each  Scenario. | | |

Table :SignUp Textual description

|  |  |  |
| --- | --- | --- |
| Use-Case name: SignUp | | |
| Description | The Actor User visits the homepage and progressively enters his personal Info to have a created account. | |
| Actors | User. | |
| Date | 02/09/2023 | |
| Stakeholder | ICES Sarl. | |
| Version | 1 | |
| Precondition | The Actor has no created account in the system. | |
| Trigger | The Actor visits the homepage through the browser. | |
| Main Course Scenario | | |
| User | System | DBMS |
| 1. This Use Case starts when the Actor visits the homepage |  |  |
| 1. Clicks on the SignUp button | 1. Display SignUp Page |  |
|  | 1. Prompts for entering personal info in the various fields |  |
| 1. Actor enters personal info and clicks on the signup button | 1. Performs Validation of data inputs using Data validation rules |  |
|  | 1. Performs MatriculeID Verification. |  |
|  | 1. Queries data to the database. |  |
|  |  | 1. Sends query results |
|  | 1. Make a Dashboard for the visitor. |  |
|  | 1. Display the Dashboard to the requesting visitor. |  |
| 1. This Use Case ends when the Actor has a Student Account and accesses their Dashboard. |  |  |
| Alternative Scenario: Invalid Credentials | | |
| User | System | DBMS |
|  | 4. a Fails to Validate data inputs. |  |
|  | 4. b Display an error message indicating the problem to be fixed. |  |
|  | 4. c Prompts for corrected credentials. |  |
| 4.d Use Case continues to step 3 of the Main Course Scenario. |  |  |
| Alternative Scenario: Failed MatriculeID Verification | | |
| Visitor | System | DBMS |
|  | 5. a Detects redundancy of MatriculeID |  |
|  | 5. b Displays error message indicating the problem to be fixed. |  |
| 5. c Use Case continues to step 3 of the Main Course Scenario. |  |  |
| Post Condition of Success | The user has a created account and access to the dashboard. | |
| Post Condition of Failure | The user has no created account. | |

Table :Login Textual Description

|  |  |  |  |
| --- | --- | --- | --- |
| Use-Case name: Login | | | |
| Description | This Use Case permits an Actor with a User account in the system to Login into the system | | |
| Actors | User | | |
| Date | 04/09/2023 | | |
| Stakeholder | ICES\_Sarl | | |
| Version | 1 | | |
| Precondition | Actor User has an account in the system | | |
| Trigger | User Clicks on the Login button | | |
| Main Course Scenario | | | |
| User | | System | DBMS |
| 1. This Use Case starts when the Actor User clicks on the Login button. | | 1. Displays Login page. |  |
|  | | 1. Prompts for User Credentials (Name and Password). |  |
| 1. Fill in fields with Credentials (Email and Password). | | 1. Queries User(email) related-data from the database. | 1. Sends query results. |
|  | | 1. Verifies User Credentials against query results. |  |
|  | | 1. Display User Dashboard. |  |
| 1. Use Case ends when the Actor User gains access to the Dashboard. | |  |  |
| Alternative Scenario: Verification Failure | | | |
| User | | System | DBMS |
|  | | 7.1 Fails to verify User specified Credentials |  |
|  | | 7.2 Displays an error message indicating the problem to be fixed |  |
| 7.3 Use Case Continues Step 2 of the Main Course Scenario | |  |  |
| Post Condition of Success | The user gains access to Personal dashboard | | |
| Post Condition of Failure | User unable to access dashboard | | |

3.) Communication Diagram

a.) Definition

A Communication diagram focuses on the messages between a group of objects and the underlying messages of the objects. They show how objects collaborate to meet a goal. In other words, they are time- and space-oriented and emphasize the overall interaction, the elements involved, and their relationships.

b.) Formalism

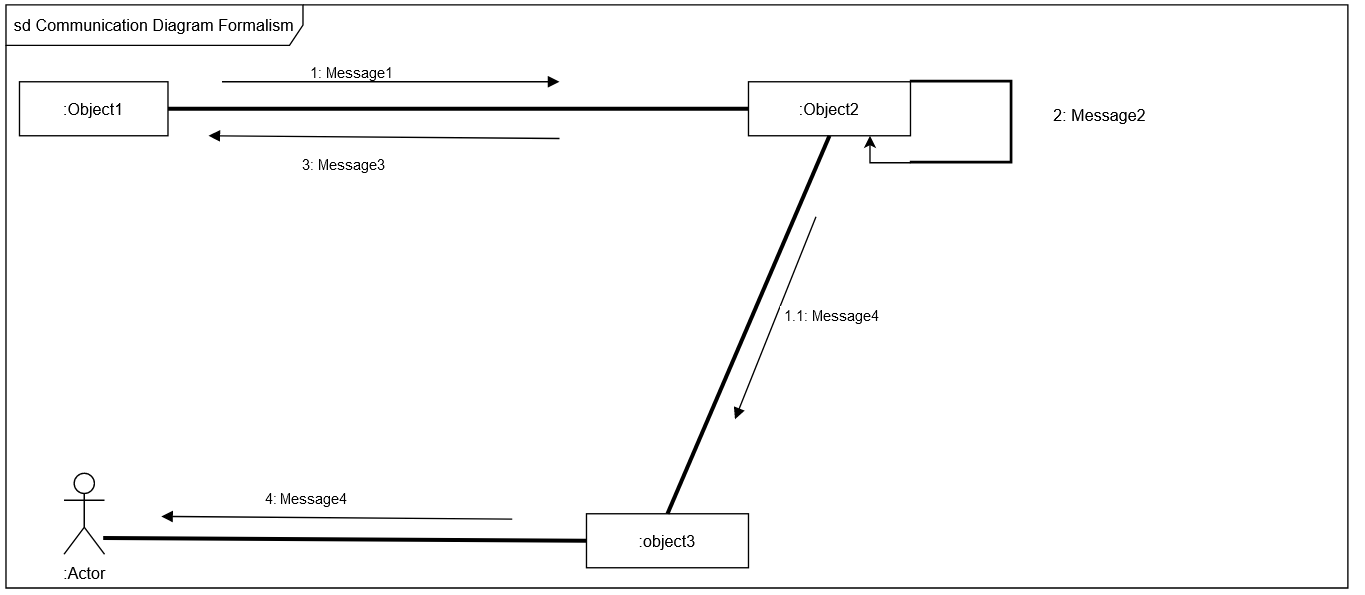


Figure : Communication Diagram Formalism

c.) Components of a Communication Diagram.

Table : Communication diagram components

|  |  |  |
| --- | --- | --- |
| ELEMENT | NOTATION | DESCRIPTION |
| Message | 1messageA()[condition] | It defines the communication between two objects at a given instant. |
| Link |  | a link is a tool in a UML diagram to indicate that two objects communicate with each other. |
| Object |  | Objects can be any useful item that has identity, structure, and  behaviour. |
| Actor |  | A role played by an entity. |

d.) System Login Communication diagram

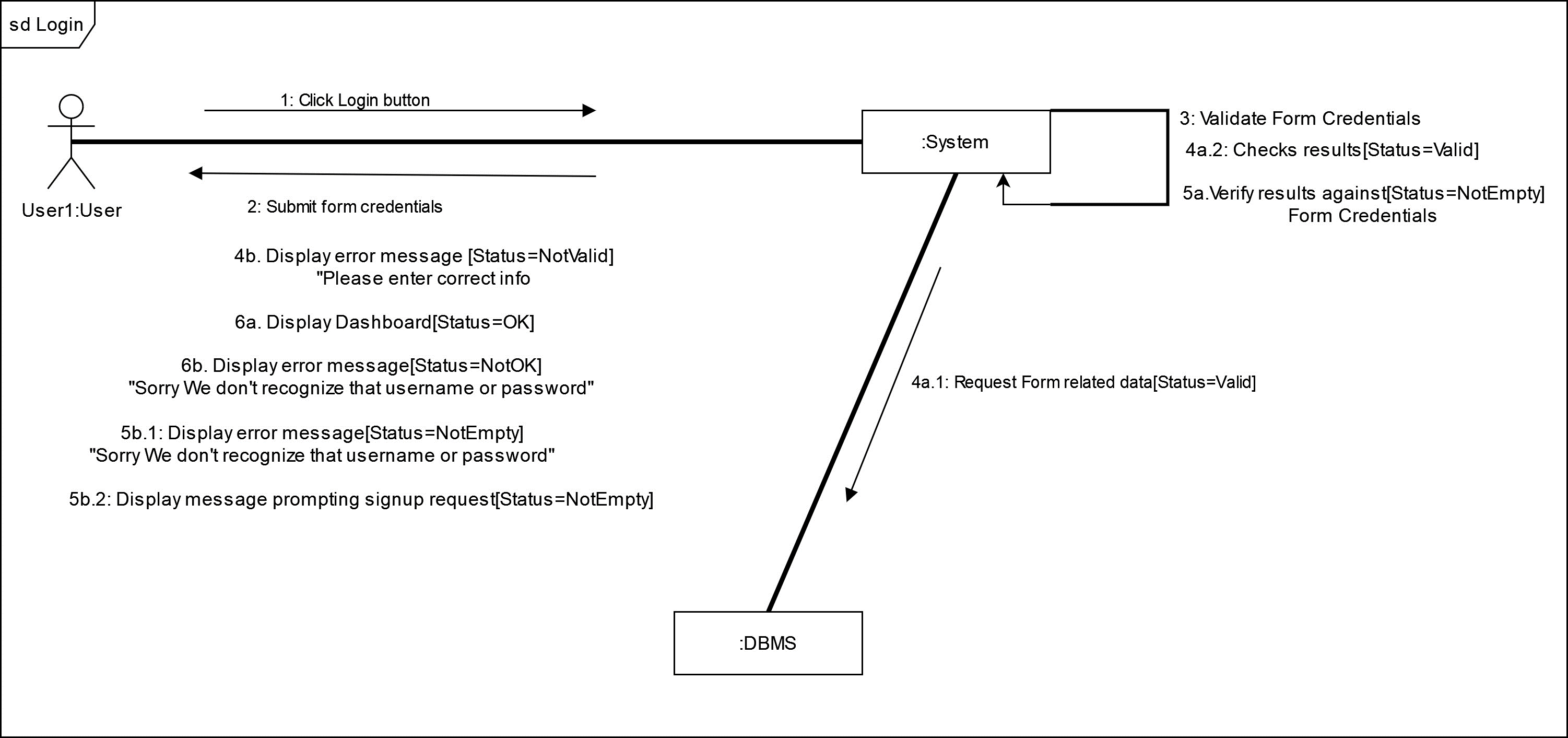


Figure :System Login Communication diagram

2.) Sequence Diagram

a.) Definition

A sequence diagram is an Interaction diagram which captures the exchange of messages between participating objects. They are time-oriented and emphasize the overall flow of an interaction.

b.) Formalism

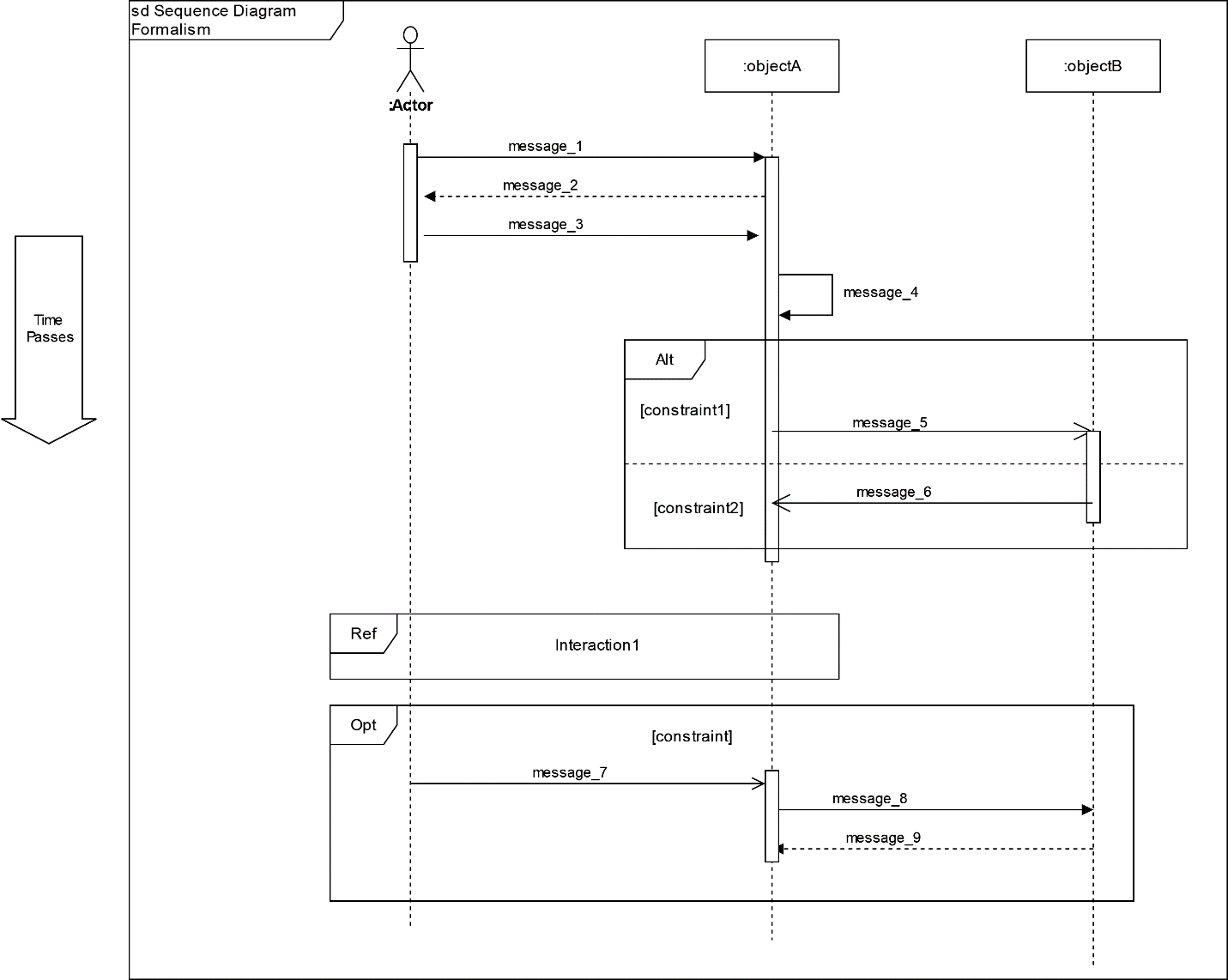


Figure : Sequence Diagram Formalism

c.) Components of a Sequence Diagram.

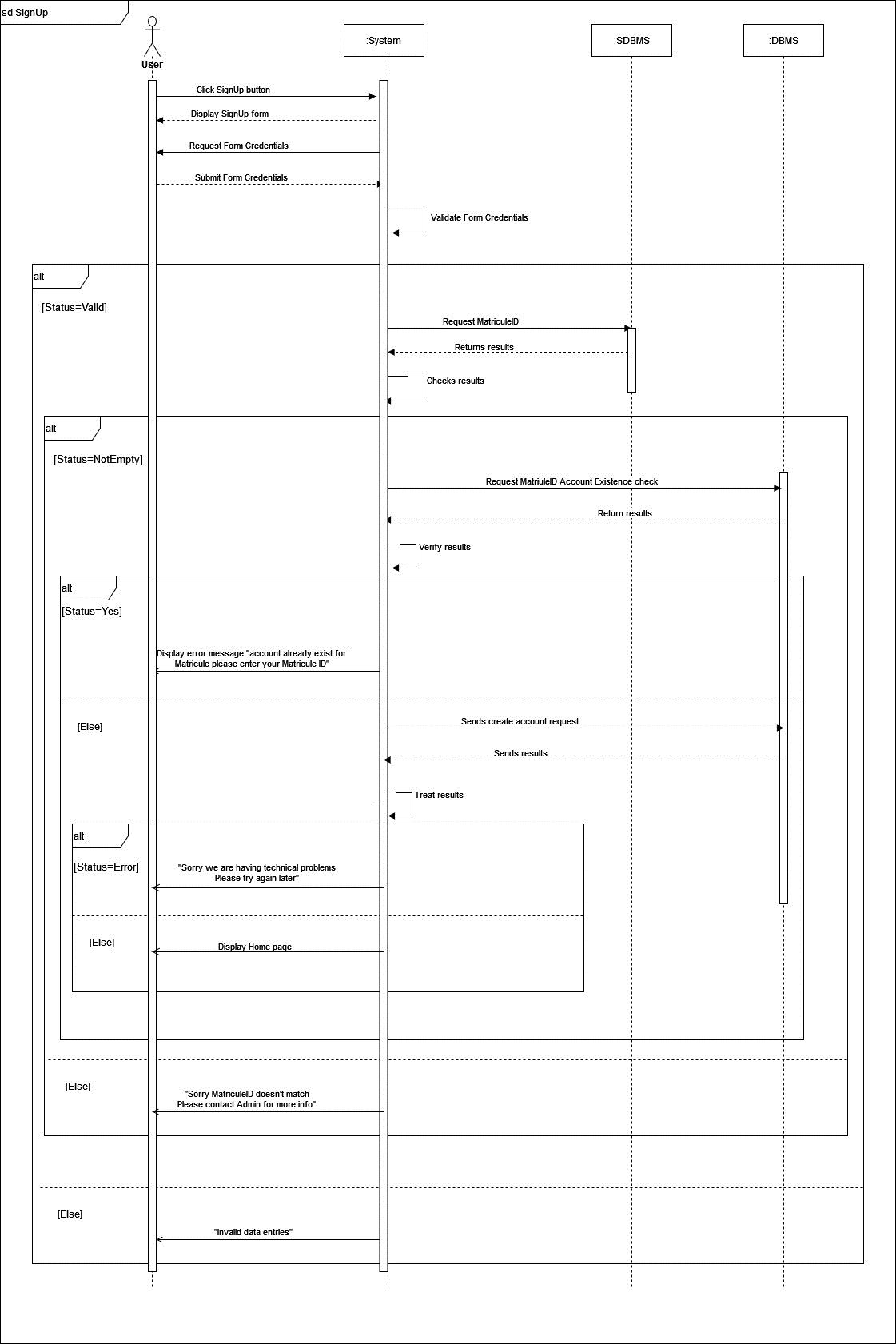
Table :Sequence diagram components

|  |  |  |
| --- | --- | --- |
| ELEMENT | NOTATION | DESCRIPTION |
| Lifelines |  | It is a vertical dashed line from an element which represents the existence of the element over time. |
| Object |  | It’s an instant of a class. |
| Actor |  | Communicate with other objects. |
| Activation |  | It represents the period during which an element is performing an operation. |
| Message |  | It indicates the communication between objects. We have Synchronous messages, reply messages, Asynchronous messages and Self messages as types of messages. |

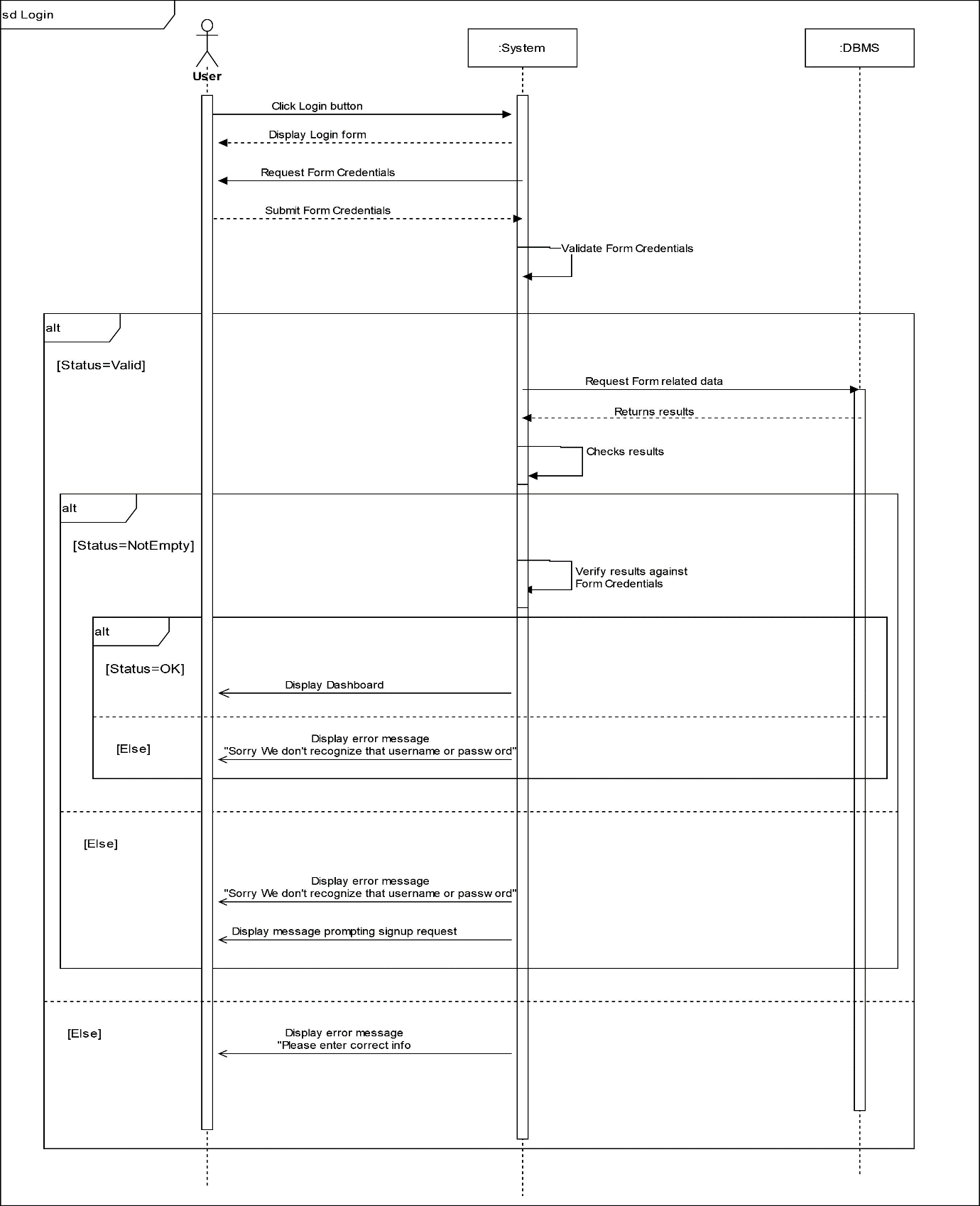
***Table 12: Components of Sequence Diagram (Continued)***

|  |  |  |
| --- | --- | --- |
| ELEMENT | NOTATION | DESCRIPTION |
| Combined interaction fragment |  | They represent an articulation of interaction. We have Alt, opt, and Ref as some combined interaction fragment types. |

d.) Sequence diagram: Signup

 Figure :Sequence Diagram:SignUp

e.)Sequence diagram: Login

 Figure :Sequence diagram: Login

2.) Activity Diagram

a.) Definition

An activity diagram is basically an extended flow chart which represents the flow from one activity to another activity. It Allows you to show the flow of behaviour across multiple classes.

b.) Formalism

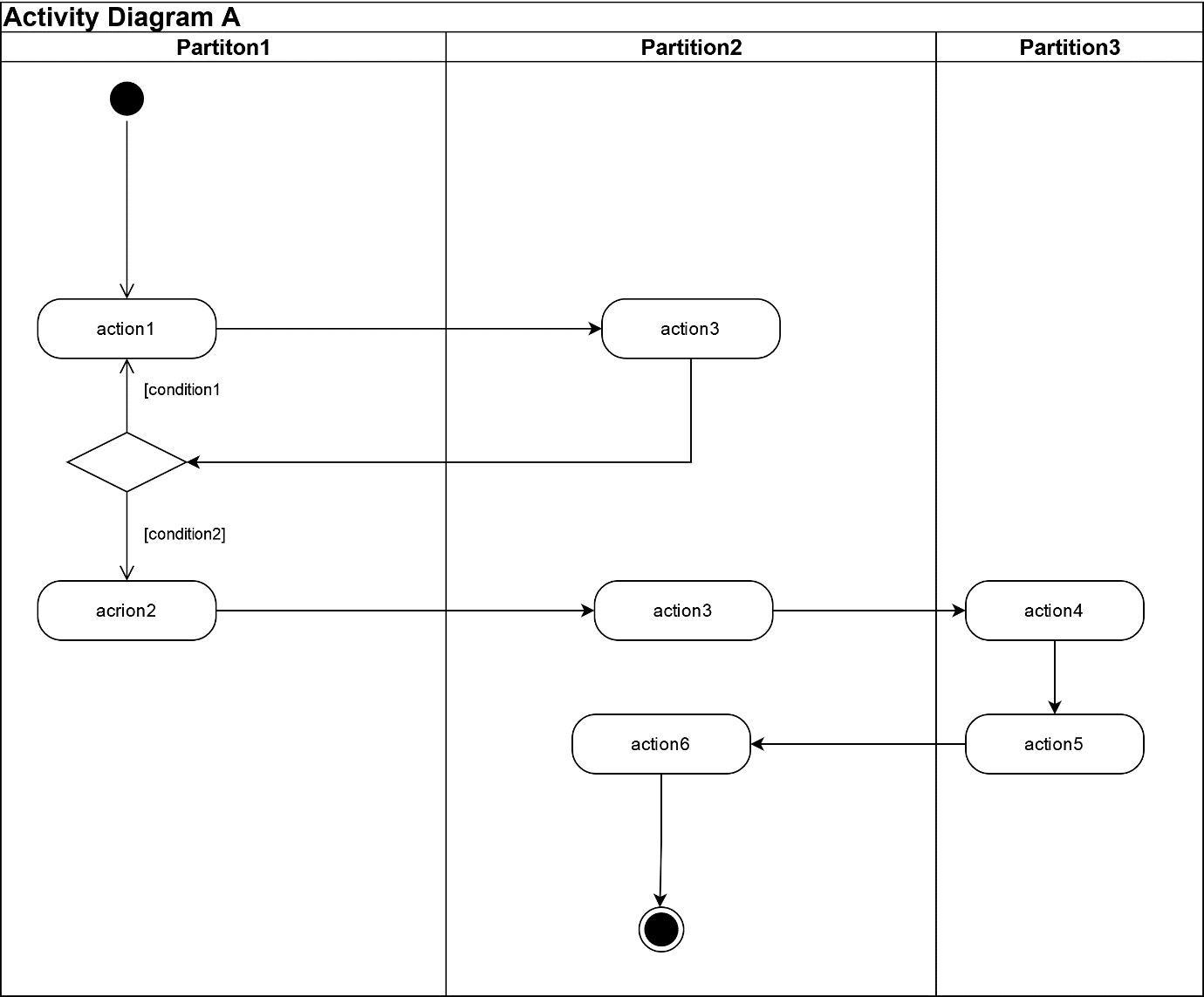


Figure :Activity diagram formalism

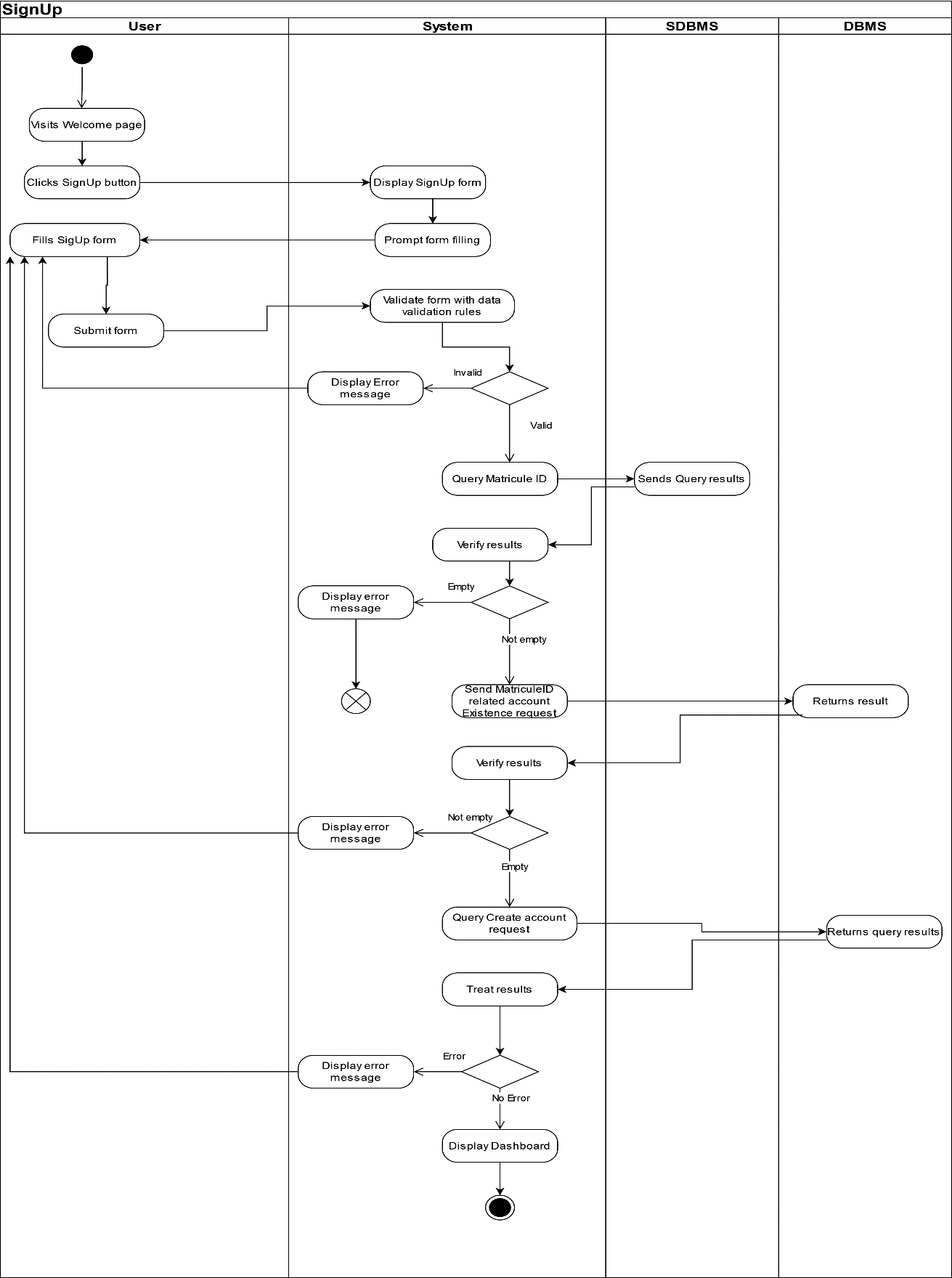
Table :Components of an Activity Diagram

|  |  |  |
| --- | --- | --- |
| ELEMENTS | NOTATION | DESCRIPTION |
| Activity |  | Activities contain sequences of actions and/or other activities. You use activities to group sequences of actions together. |
| Action |  | An action is a simple piece of behaviour an action. |
| Object node |  | Represents an activity  node that indicates an instance of a particular classifier in the activity. |
| Control flow |  | Connects actions and activities together; shows the sequence of execution. |
| Initial node |  | This shows the starting  point or first activity of the flow. |
| Final activity node |  | Ends all control flows and object flows in an activity, using the final-activity node. |
| Final flow node |  | End some — but not all — flows inside an activity. |

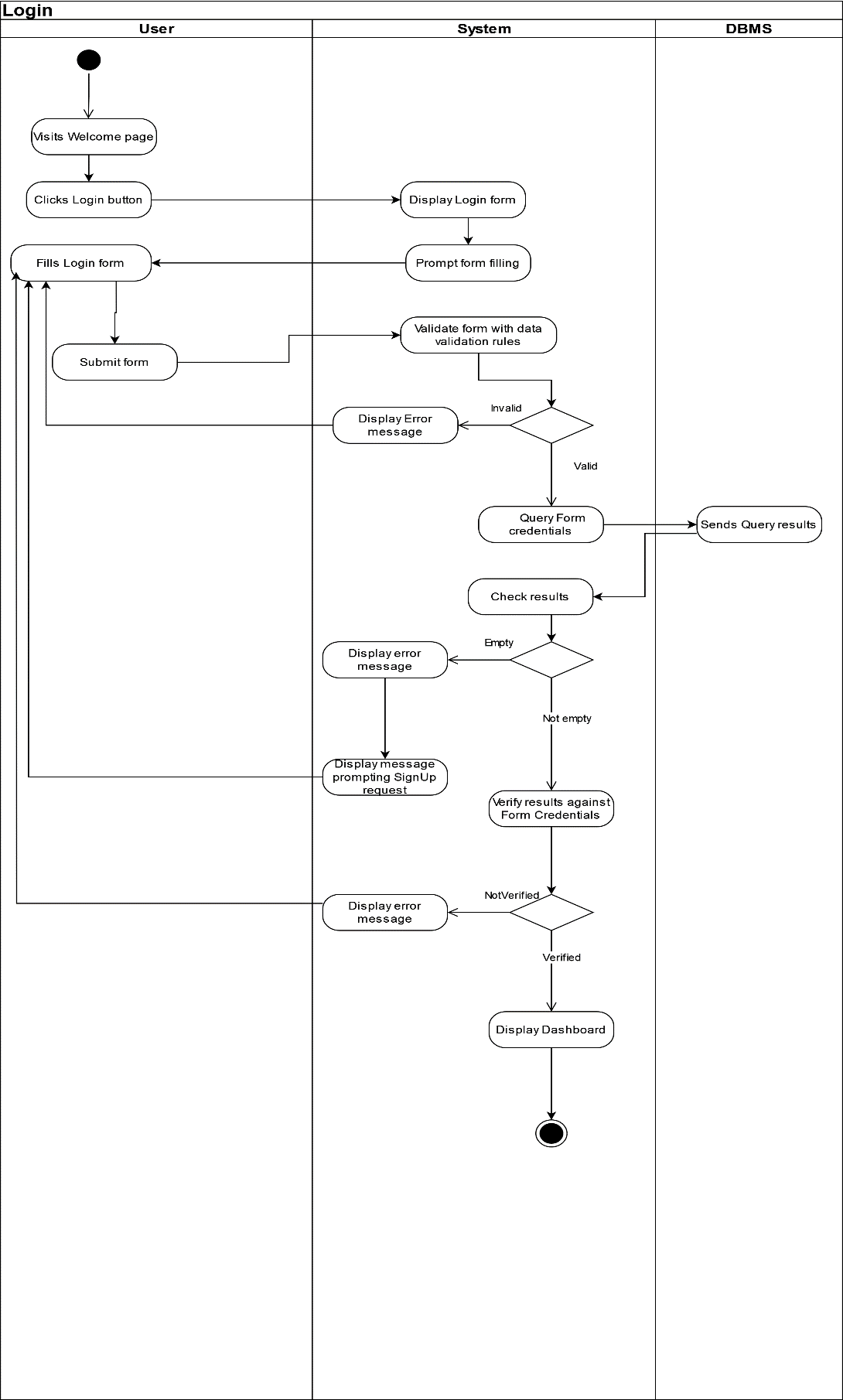
***Table 12: Components of Activity Diagram (Continued)***

|  |  |  |
| --- | --- | --- |
| ELEMENT | NOTATION | DESCRIPTION |
| Decision node |  | A decision node uses a test to make sure that an object or control flow goes down only one path. |
| Merge |  | Brings separate decision paths back together. |
| Fork node |  | Used to split behaviour into concurrent operations. |
| Join node |  | Used to synchronize incoming concurrent  Flows. |
| Swimlane |  | Swimlane is a visual region  in an activity diagram that indicates the element that has responsibility for action states within the region. |

c.) SignUp Activity Diagram

Figure : SignUp Activity Diagram

c.) Login Activity Diagram

Figure : Login Activity Diagram

CONCLUSION

The Analysis phase has enabled us to make an inventory of the functional needs. While sticking to the 2TUP method, we initially enumerate the various choices and needs necessary for our system study and finally describe the interaction between actors, and objects. Next is the Conception Phase where we present the 2 branches of the 2TUP with respect to our system.

PART IV

CONCEPTION PHASE

Preface

The Conception phase is the part which makes it possible to capture the technical needs and establish the design and architecture of our system. It links the realization phase with the analysis phase.

Contents

1. INTRODUCTION

2. TECHNICAL BRANCH

2.1 CAPTURE OF TECHNICAL NEEDS

2.2 GENERIC DESIGN

3. IMPLEMENTATION BRANCH

3.1 PRELIMINARY DESIGN

3.2 DETAILED DESIGN

INTRODUCTION

The capture of the technical needs can be carried out as soon as one knows which materials will be used. Materials include more precisely machines, networks, components etc. At this level, we will examine some diagrams such as class diagrams, package diagram, state machine diagrams etc.

2. TECHNICAL BRANCH

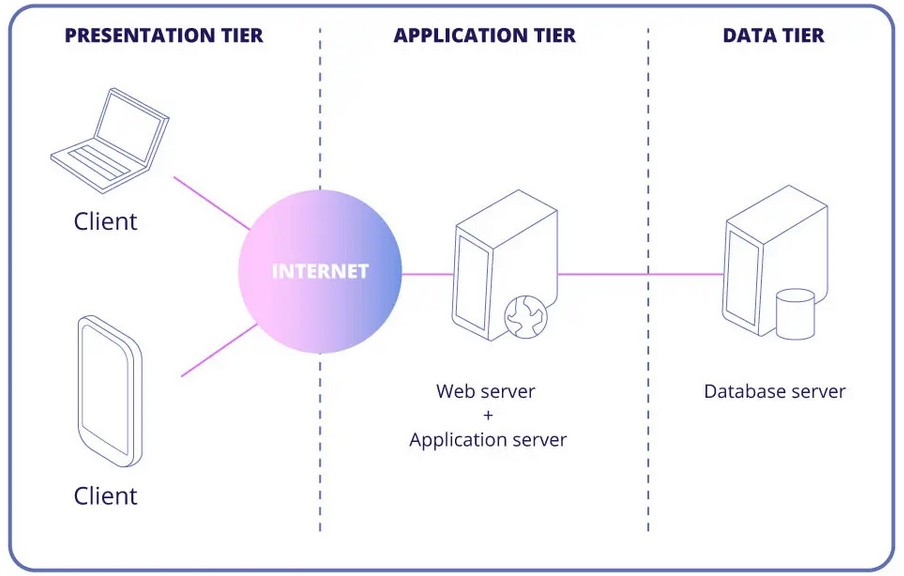
The technical branch captures the technical needs and proposes a generic design validated by a prototype.

2.1) CAPTURE OF TECHNICAL NEEDS

a.) Physical Architecture

Modern web application architecture typically consists of several layers that work together to provide a robust, scalable, and maintainable system. Each layer serves a specific purpose, and together they form a cohesive and integrated solution that can handle a variety of tasks. Choosing the right architecture is a complex decision. While scalability, maintainability, and Data Integrity are essential considerations, other factors such as budget, time-to-market, and technology stack also come into play reason why we choose Three-tier architecture. **Three-tier architecture** is a well-established software application architecture that organizes applications into three logical and physical computing tiers. The three-tier architecture is the most popular implementation of a multi-tier architecture and consists;

* Presentation tier, which is the user interface and communication layer of the application or platform, where the end-user interacts with the application or platform.
* Application tier, also known as the logic tier collects information from the presentation tier process sometimes against other information in the data tier - using business logic, a specific set of business rules.
* Data tier which represents our DBMS.

Figure :3-Tier architecture(Source: https://mobidev.biz/blog/web-application-architecture-types)

b.) Logical architecture

Model View controller or MVC as it is popularly called, is a software design patten for developing application. A model view controller patten 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.

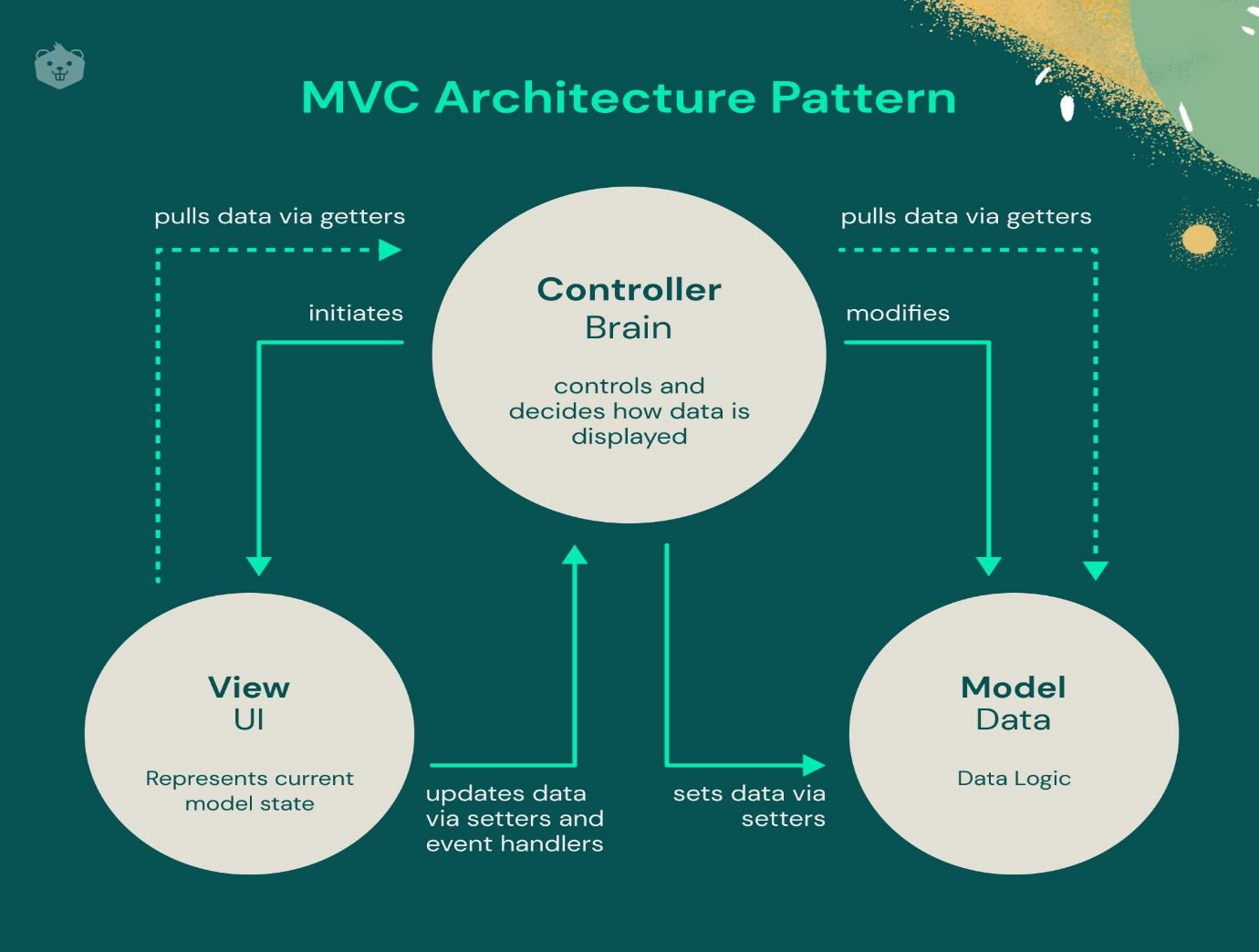


Figure :MVC Architecture (Source: https://www.crio.do/blog/understand-mvc-architecture/)

2.2 GENERIC DESIGN

a.) Hardware diagram of the system

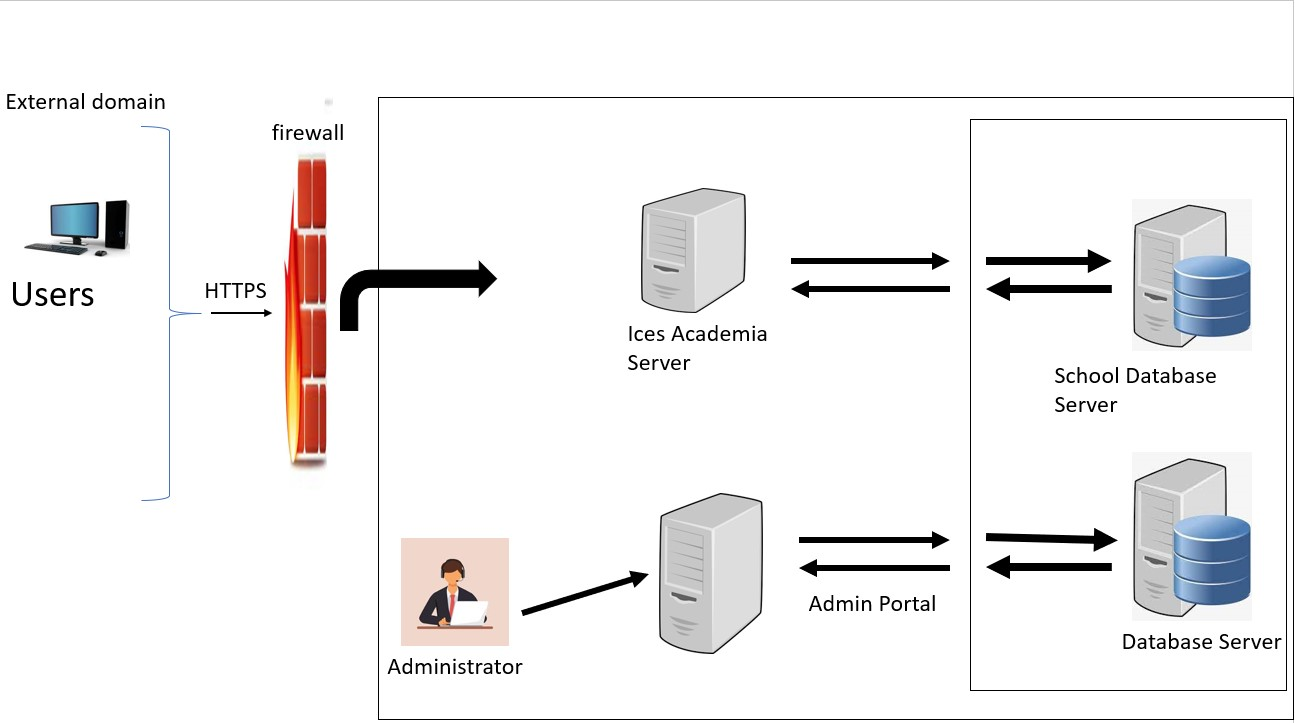
The hardware diagram simply shows how the system components of our system are deployed; it shows the positioning of each device into it right proportion.

Figure :Hardware Diagram

b.) High Level Architectural Diagram of the Software

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



Figure :High Level Architecture

3.) IMPLEMENTATION BRANCH

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

3.1) PRELIMINARY DESIGN

1.) State Machine Diagram

a.) Definition

It defines the different states of anobject during its lifetime and these states are changed by events. Since it records the dynamic view if a system, it portraits the behaviour of a software application.

b.) Formalism

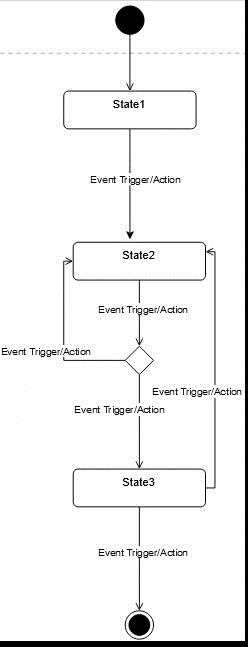


Figure :State Machine Diagram formalism

c.) Components of a State Machine Diagram

Table :State Machine Diagram Components

|  |  |  |
| --- | --- | --- |
| Element | Diagram Relationship | Description |
| State |  | Models a situation during which a certain invariant condition holds. |
| First (Initial) State |  | It defines the initial state |
| Final State |  | It represents the final state or the end of a system. |
| Transition |  | It is a change of control from one state to another due to the occurrence of  some events. |
| 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. |

d.) System Login State Machine Diagram

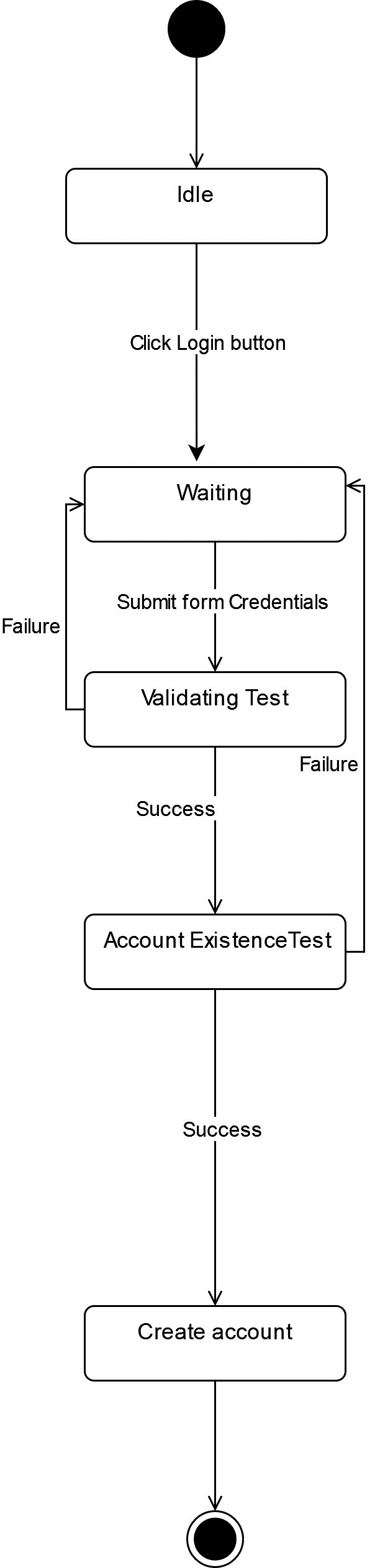


Figure :Login State Machine Diagram

d.) System SignUp State Machine Diagram

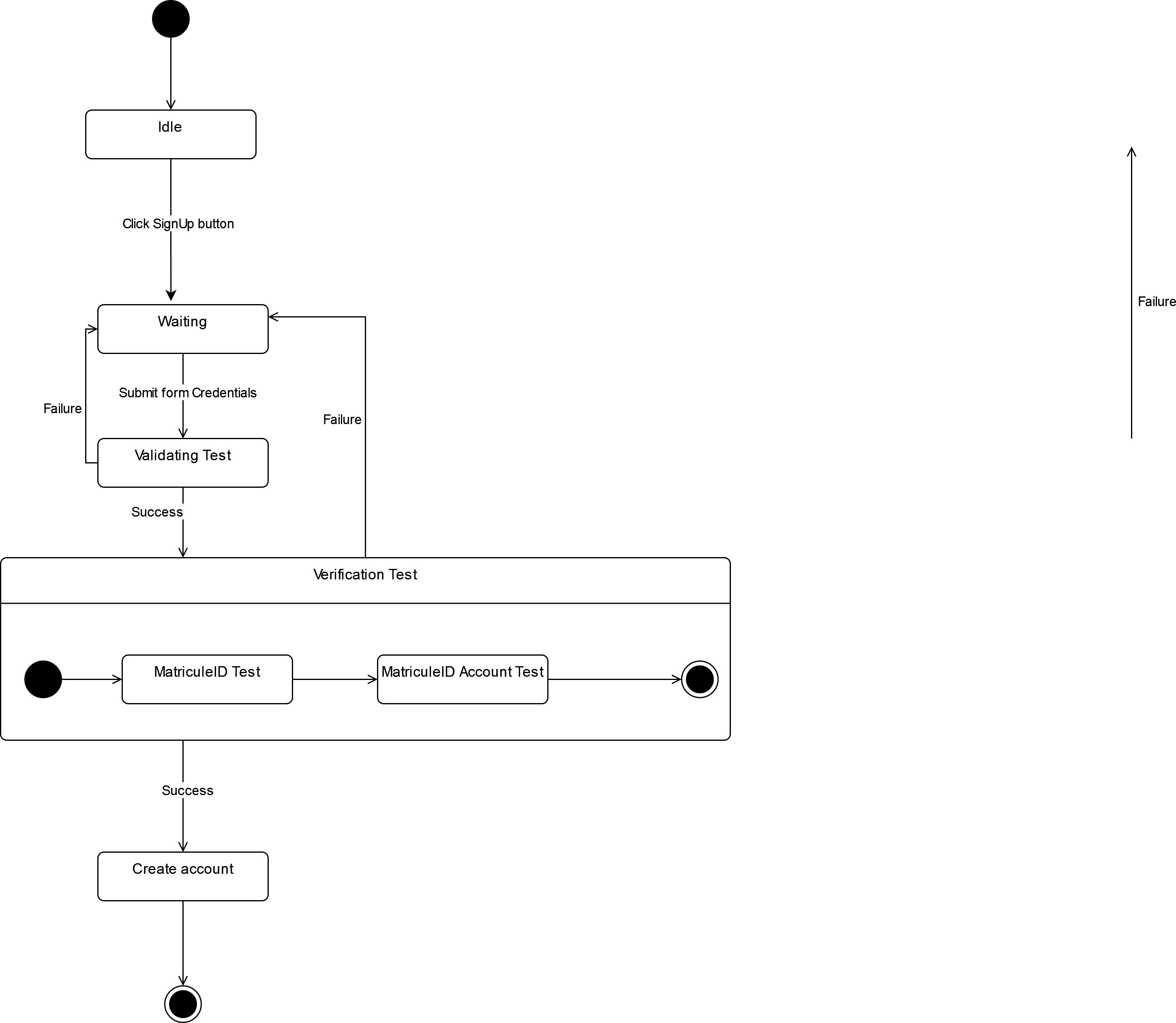


Figure :System SignUp State Machine Diagram

2.) Package Diagram

a.) Definition

This is a structural diagram used to show the organisation and arrangement of various

model elements in the form of packages.

b.) Formalism

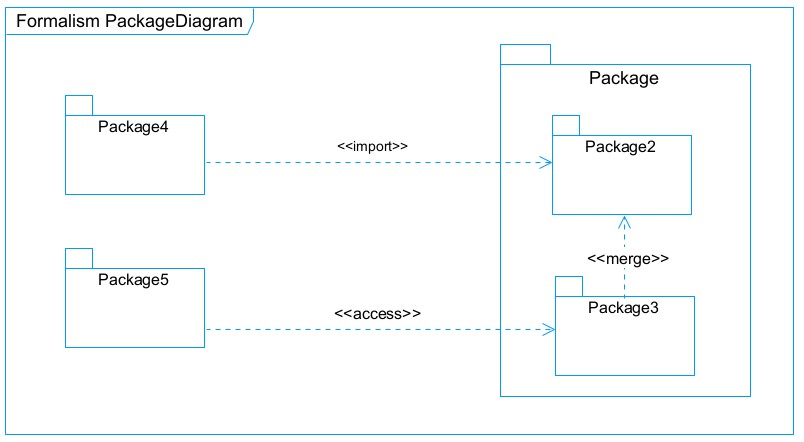


Figure :Package Diagram Formalism

c.) Components of a Package Diagram

Table :Components of a Package diagram

|  |  |  |
| --- | --- | --- |
| NAME | Representation | Description |
| Package |  | A package is used to group together logically created elements within a system. |
| Package import |  | It indicates that a functionality has been imported from one packet  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. |

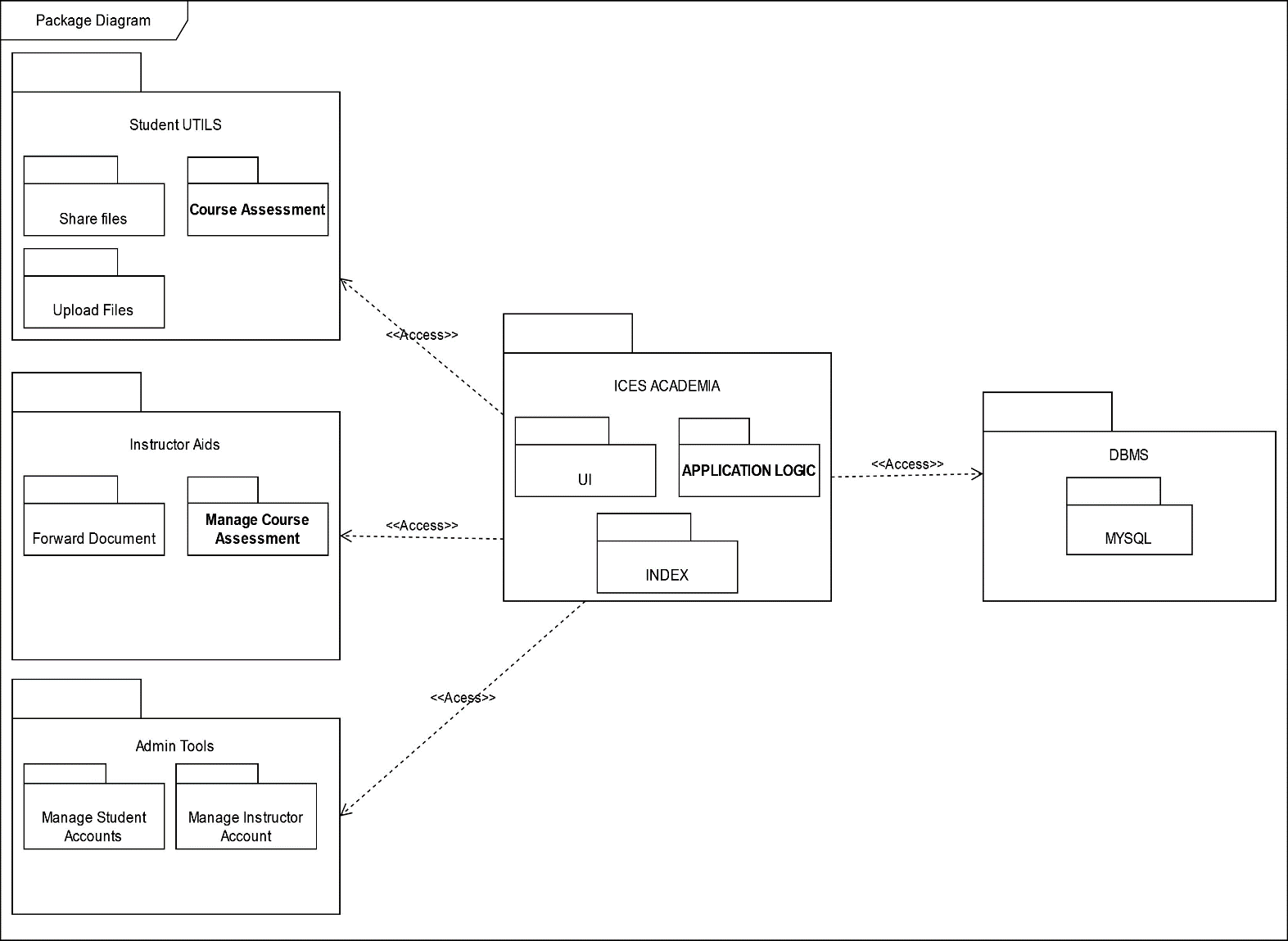
d.) Web Package Diagram

Figure :ICES ACADEMIA Web Package diagram.

3.2) DETAILED DESIGN

1.) Class Diagram

a.) Definition

Class diagram is UML structure diagram which shows structure of the designed system at the level of classes and interfaces, shows their features, constraints and relationships - associations, generalizations, dependencies**.**

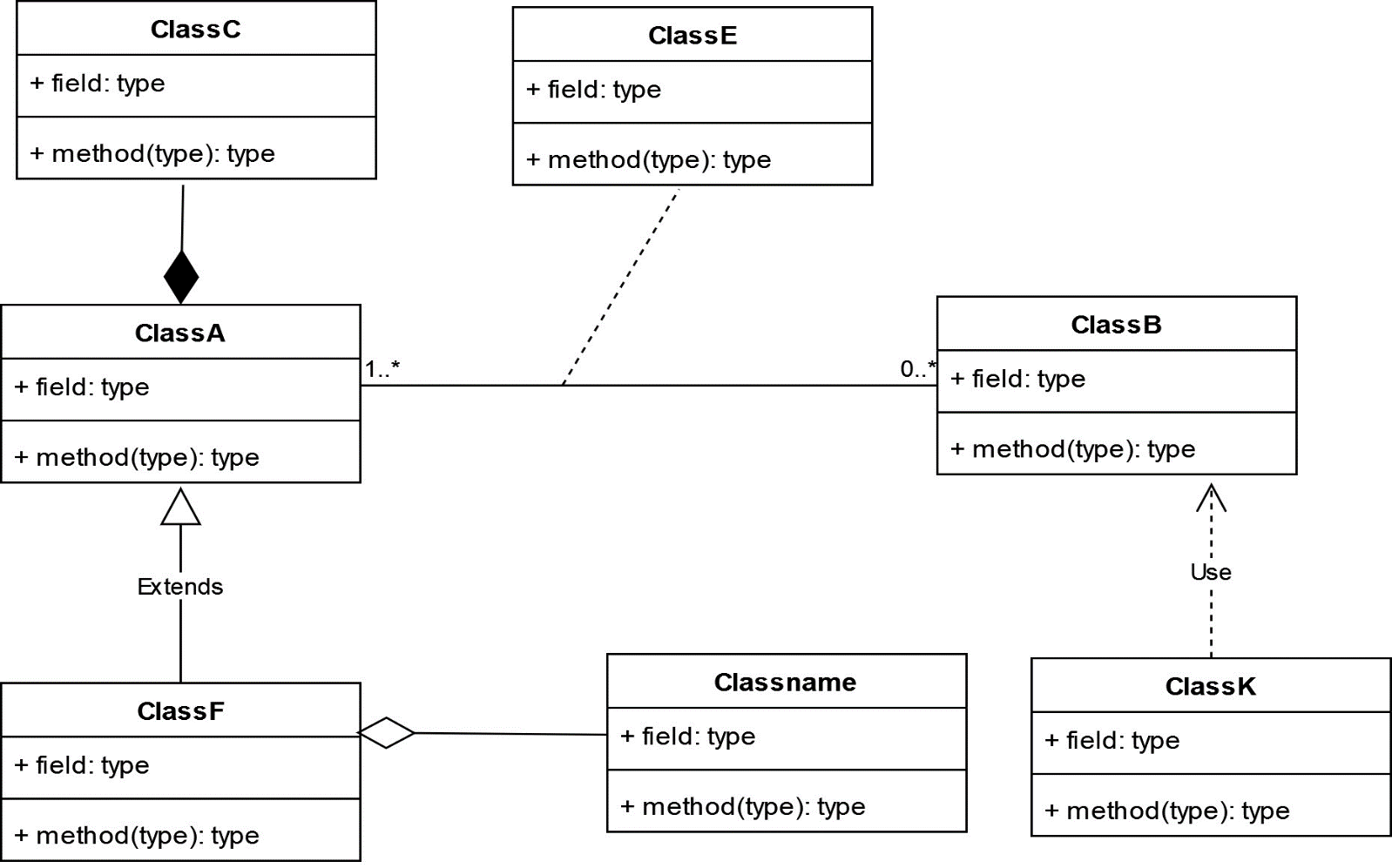
b.) Formalism

Figure :Class Diagram Formalism

c.) Components of a Class diagram

Table :Components of a Class diagram

|  |  |  |
| --- | --- | --- |
| Element | Representation | Description |
| class |  | It defines the structure, the behaviour and the relationship of these objects. |
| Composition |  | If a parent of a composite is deleted, usually, all of its parts are deleted with it. |
| Aggregation |  | It models the notion that one object uses another object without "owning" it and thus is not responsible for its  creation or destruction. |
| Dependency |  | It existed between two classes, if one changes it may cause the change in the order, but the other way around. |
| Generalization |  | It is a relationship between a whole thing (called superclass) and a more specific thing (called subclass) |
| Association |  | It is a relationship between 2 or more classes. Types of association include Binary, Nary associations. |
| Association class |  | It is a class formed between 2 or more countries. |

d.) ICES ACADEMY Class diagram

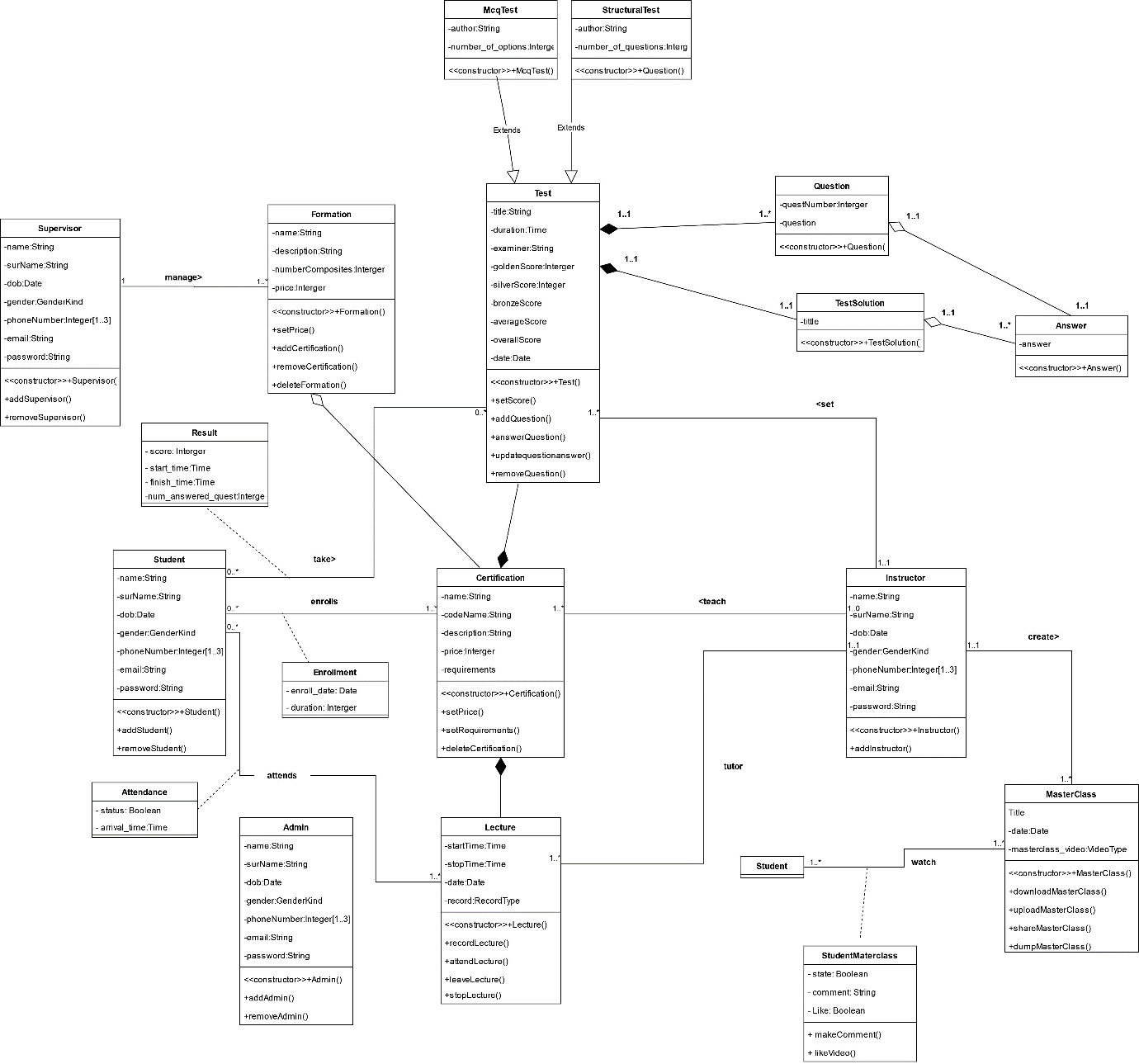


Figure : ICES ACADEMY Class diagram

CONCLUSION

We planned the structure and deployment of our system in this chapter. We considered the technical constraints, components, deployment targets, and interactions of our system. This gave us a clear overview of how our system will work and what are the main challenges and opportunities. In the next chapter, we will implement our system using various technologies, tools, and methods. We will also show some examples of the system’s functionalities and performance.

PART V

**REALISATION PHASE**

Preface

In this phase we will to straight forward in the implementation of our solution, we will base

ourselves on the analysis and conception phases.

Contents

1. INTRODUCTION

2. DEPLOYMENT DIAGRAM

3.COMPONENT DIAGRAM

4. PRESENTATION OF DEVELOPMENT TOOLS

5. PRESENTATION OF LANGUAGE USED

6.CONCLUSION

INTRODUCTION

In this project, we aim to develop a system that solves a specific problem or meets a certain need. To achieve this goal, we have followed a series of steps that include analysis, conception, and realization. In the analysis phase, we defined the problem and the requirements of the system. In the conception phase, we designed the architecture and the components of the system. In this report, we will focus on the realization phase, where we will implement our solution using various technologies. We will also explain how we used the entity relational diagram to model the relationship between the entities of our system. This phase is as critical as the previous ones, as it will determine the functionality and performance of our system.

2.) DEPLOYMENT DIAGRAM

a.) Definition

A Deployment diagram models the hardware components use to implement the system

and the association between these components.

b.) Formalism

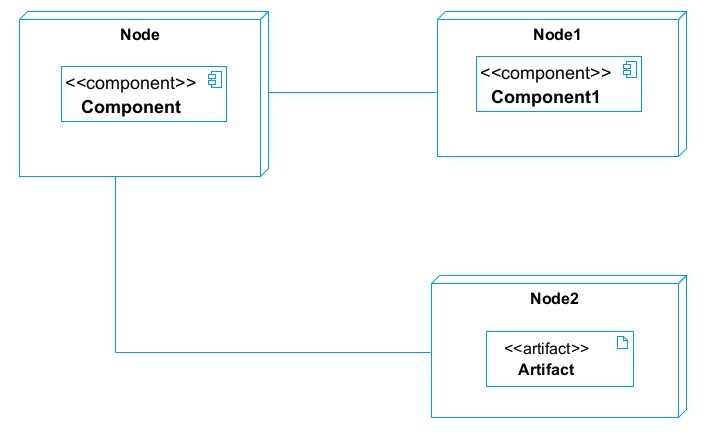


Figure :Components of a Deployment Diagram

c.) Components of a Deployment Diagram

Table : Deployment Diagram components

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

d.) System Deployment diagram

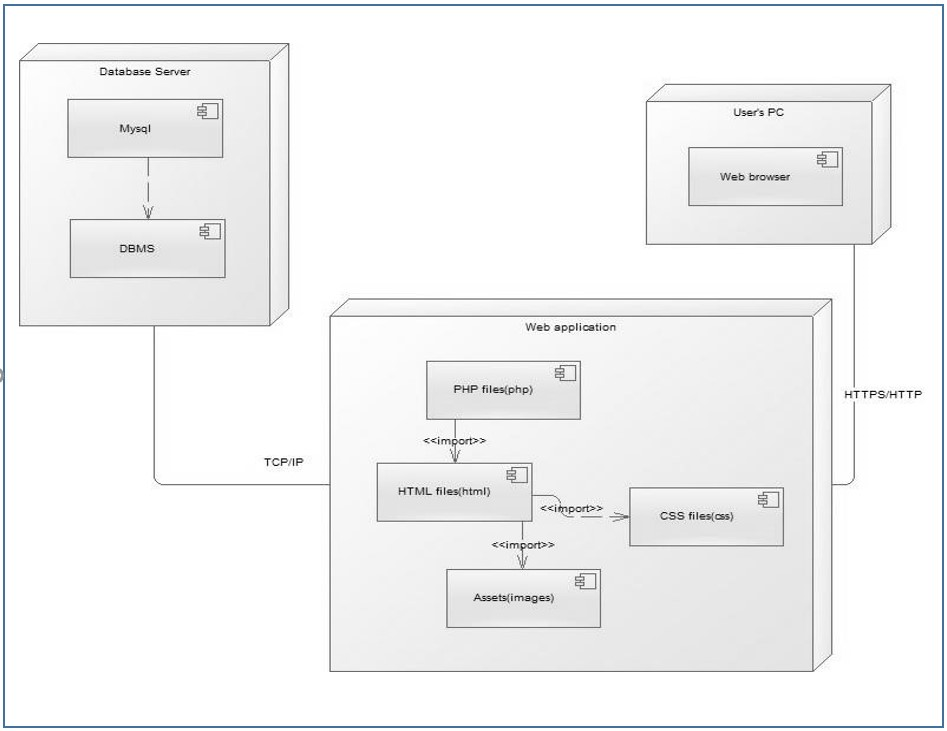


Figure :ICES ACADEMY Deployment diagram

3.) 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

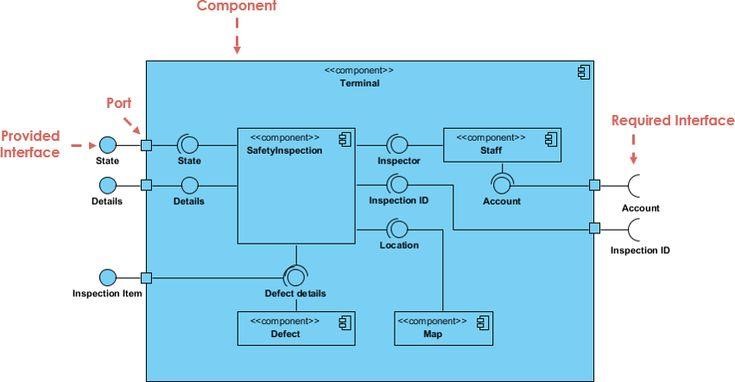


Figure :Formalism of a Component Diagram

c.) Components of a Deployment Diagram

Table :component of component diagram

|  |  |  |
| --- | --- | --- |
| Elements | Notation | Description |
| Component |  | Represented as a rectangle with the component’s name inside. Often it also has an icon of two smaller rectangles on its left side. Symbolizes a modular part of a system which can encapsulate certain functionality or a group of functionalities. |
| Interface |  | Depicted as a circle (or lollipop representation) and sometimes as a semi-circle attached to a component.  Defines a set of operations which specify the responsibility of a component. |
| Dependency |  | Illustrated dashed arrow. Signifies that one component depends on other components to function properly. |
| Port |  | Represented as a small square on the edge of a component. Defines an entry or exit point from the component for data or control flow. |
| Connector |  | Shown as a solid line between two components or ports. Signifies the communication path between components. |

d.) System Deployment diagram

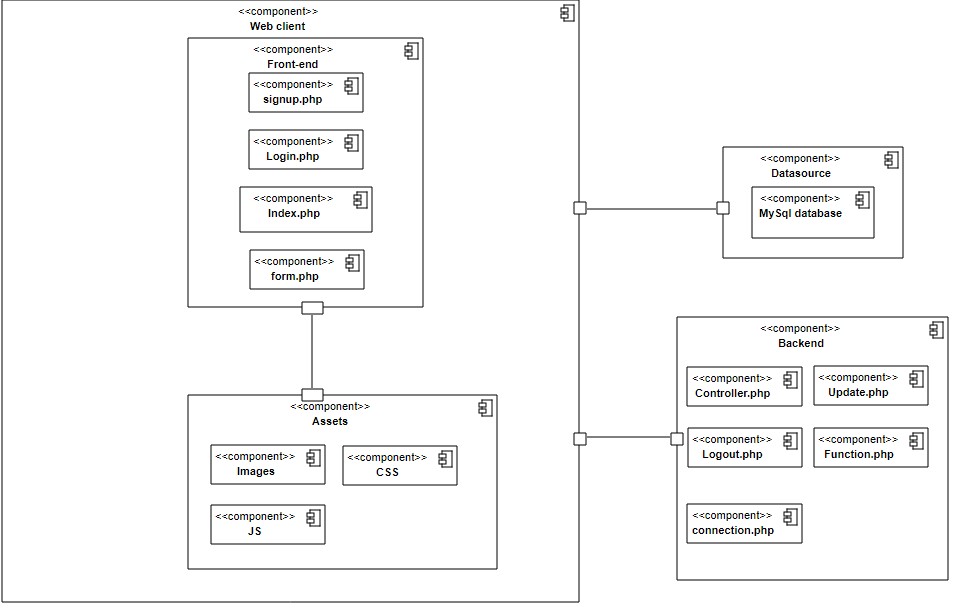


Figure :Component diagram of our website

4.) PRESENTATION OF THE TOOL USED

a.) SOFTWARE RESOURCES

Table :Software resources used in project

|  |  |  |
| --- | --- | --- |
| Software | Logo | Roles |
| Windows 11 OS Home |  | The software where all was done that is management of the computer resources |
| Browser |  | Used in doing researches |
| StarUML | A colorful star with black text  Description automatically generated | Used in drawing UML diagrams |
| Gantt Project |  | Used to do planning of the realization of the work and to present the schedule of the work |
| Microsoft word |  | Software for the treatment of text, used in drafting the report |
| Visual studio | A blue ribbon with a cross  Description automatically generated | IDE or a studio to write the code of the application both front end and backend |

a.) HARDWARE RESOURCES

Table :Hardware Resources

|  |  |
| --- | --- |
| Hardware | Notation |
| Memory 12GB RAM |  |
| HARD DISK 1To |  |

### 

5.) PRESENTATION OF THE LANGUAGE USED

Table :Language used in project

|  |  |
| --- | --- |
| Software | Roles |
| HTML | A standard markup language for creating web  Pages. |
| CSS | A style sheet language used for describing the  presentation of a document written in a  markup language. |
| PHP | A service-side scripting language that is used  to develop static or dynamic websites or web  applications. |
| SQL | A special-purpose programming language  designed for managing data held in a  relational database management system, or  for stream processing in a relational data  stream management system |
| UML | A modeling language used for the modeling of project diagrams. |

CONCLUSION

Having arrived at the end of our project, we can say with absolute certainty that the internship was extremely beneficial to us. This is because in the course of the internship we faced many challenges hence pushing us to do researches which made us acquire knowledge.

PART VI

FUNCTIONALITY TEST

Preface

The functionality test phase is an important part of the software development life cycle. It aims to verify that the software meets the specified requirements and performs the expected functions. It also helps to identify and fix any functionality-based errors or defects that may affect the user experience or the quality of the software.

Contents

1. INTRODUCTION

2. PRESENTATION TESTING METHOD

3. SYSTEM TESTING

4.CONCLUSION

INTRODUCTION

In these phase we will look at the different testing methods and the System Testing.

2.) PRESENTATION OF TESTING METHOD

Functionality test is a type of software testing that verifies that the software performs the functions that it is intended to do. It checks whether the software meets the functional requirements and specifications that are defined by the stakeholders.

Functional testing methods are the techniques or approaches that are used to perform functional testing.

Some of the common functional testing methods are given in the table below:

Table :Testing Methods

|  |  |
| --- | --- |
| Method | Description |
| Unit Testing | Unit testing is a type of software testing that focuses on individual units or components of a software system. Example of tool is JUNIT |
| Integration Testing | Integration testing is a type of software testing that checks how different modules or components of a software system work together. |

3.) SYSTEM TESTING

a.) DB Testing

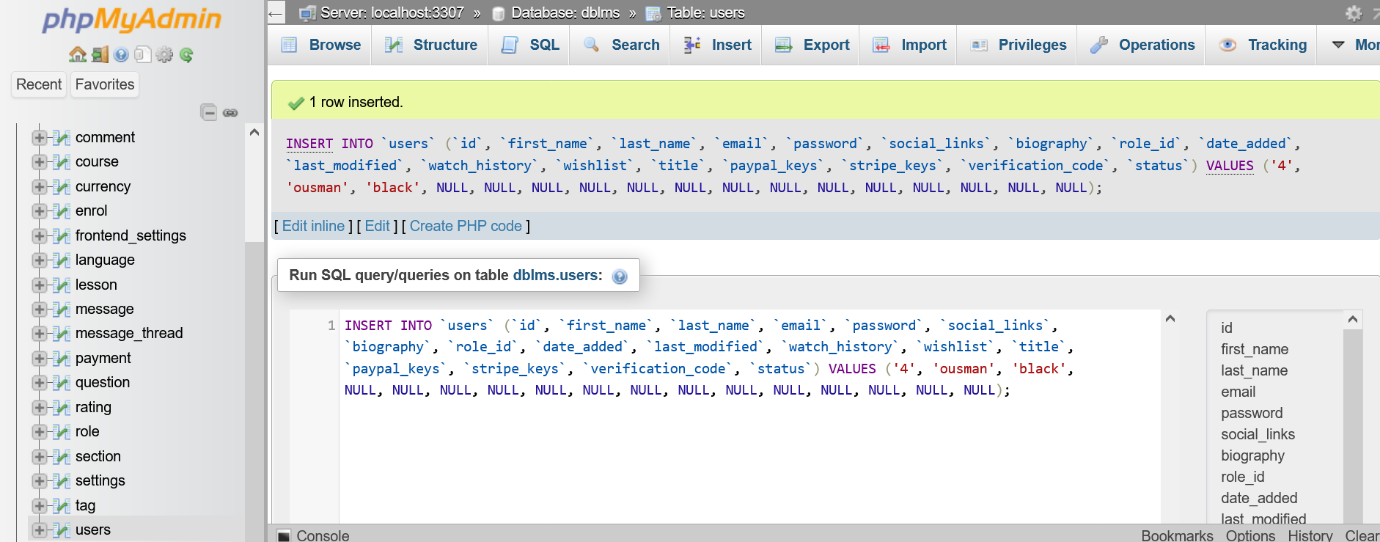


Figure :view of DB testing

PART VII

USER GUIDE

Preface

Once a solution is analysed, conceived, and deployed, it is now necessary to produce a guide for its users. This part of our document concerns the installation of the necessary tools to set up an environment for the platform in local and the user guide.

Contents

# INTRODUCTION

1. INSTALLATION GUIDE
2. USER GUIDE

CONCLUSION

INTRODUCTION

The user manual is the document created after the implementation of a software or platform or application. It answers the questions “how to use the software application or platform that is front of us?”. It is therefore crucial for us to offer the different users of our platform a guide allowing them to easily perform various operation on our platform. To achieve our goals, it will be necessary for us to use FIRE FOX web browser XAMPP SERVER all this will be done in windows 10 operating system.

I.) INSTALLATION GUIDE

After developing an application, there is a set of activities which must be carried out in order to make the system available for us. The step will mainly consist of hosting the web application we have on an application server and running the database on a database server. The following files will be used for the deployment.

Necessary Environment

* The database Server (MySQL)
* The application server (Apache)
* A Browser

a.) XAMPP SERVER INSTALLATION

XAMPP is a free and open source cross-platform we server package developed by Apache friends, consisting mainly of the Apache HTTP server; MarialDB database on a database and interpreters for scripts written in the PHP and Perl programming languages.



Figure :Xampp Server Installation

1. download the package from the [official Apache Friends webpage](https://www.apachefriends.org/index.html)
2. Open the Apache Friends websites.
3. Click the **XAMP for Windows** button to save the file on your desktop.
4. Make the installation package executable
5. Open the terminal (**Ctrl**+**Alt**+**T**)
6. Move into the folder where your installation package is located(By default, the system stores it in the **Downloads** folder.).
7. Navigate to Downloads using the command:
8. Cd /home/[username]/Downloads
9. Make the file executable by running a **chmod** command:
10. sudo chmod 755 [package\_name]
11. To make the latest XAMPP installation package executable, use the command:
12. sudo chmod 755 xampp-linux-x64-7.3.5.1-installer.run
13. Launch setup wizard sudo ./xampp-linux-x64-7.3.5.1-installer.run

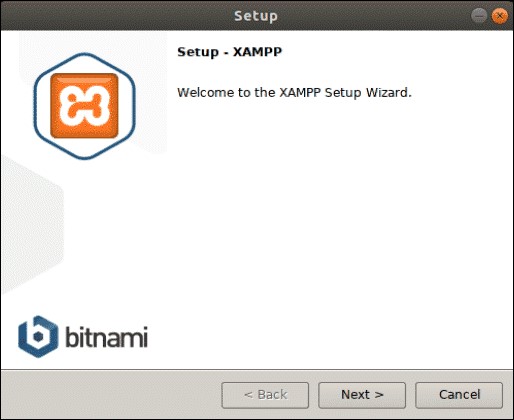
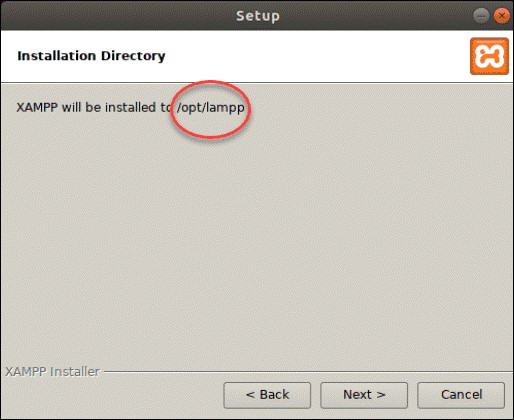


Figure :Apache setup wizard welcome page

1. Choose the components you want to install and click **Next**

Figure :Apache component installation

1. After selecting the components, the setup wizard will show you the location where it will install the software. To proceed, click **Next**.

 Figure :installation directory selection



1. The following dialogue box offers to install sponsored applications on top of the XAMPP installation. These include packages such as WordPress, Joomla, Drupal, and others. You can deny installing additional software by unchecking the **Learn more about Bitnami for XAMPP** box.



Figure :Installation Sponsor

1. Next, the wizard will notify you that it is **ready to install XAMPP** on your system. Click **Next** to start.

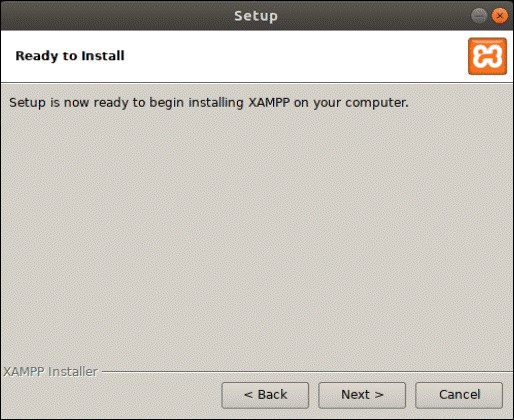


Figure :Ready to install

1. This will launch the installation process and a dialogue box showing the progress will appear on your screen.



Figure :installation progress

1. The final dialogue box should display that the setup has finished installing. You can complete the process and launch XAMPP by clicking **Finish**.



Figure :installation end

1. By clicking **Finish** in the previous step, XAMPP launches its control panel that will appear as in the image below

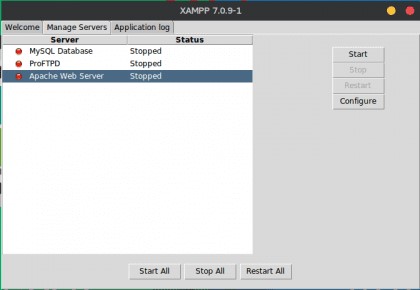


Figure :installation completed

Xampp folder, extract the compressed application file in the the htdocs folder found in the xampp folder. After this, your application can be accessed on the browser via the localhost, and we get the default index page of our application.

USER GUIDE

A user guide is a document that explains how to use a product, service, or system.

Launching the Application

Once the application is successfully deployed, you will need a browser to start the web application. The interface that will be displayed will allow you to authenticate. To access this page, simply enter the following URL

[**http://localhost/Online\_Learning\_Management\_System/**](http://localhost/Online_Learning_Management_System/)and the welcome page is displayed.

HOME PAGE VIEW

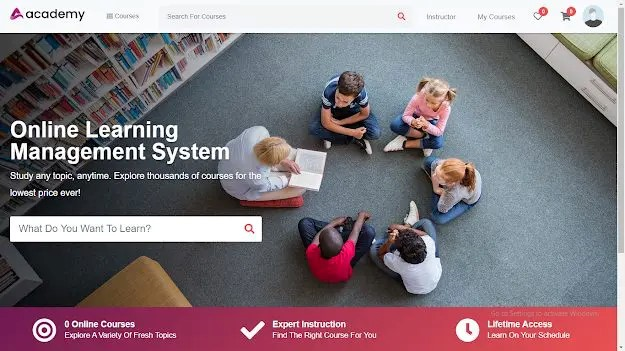


Figure :Home page view

ADMIN PAGE VIEW

This page displays the admin interface as seen below;

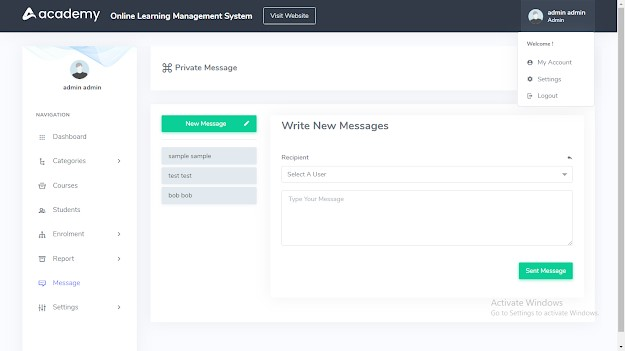


Figure :ADMIN PAGE VIEW

CONCLUSION

Having put in place the platform, it was not sufficient for we had to produce a manual that will help its various users. That is why we presented the different tools to be installed and how they are to be installed in order to run this application without any problem and how the users will use this platform once the environment is set up.

GENERAL CONCLUSION

This internship was a valuable opportunity for us to apply our computer science knowledge and skills in a real-world setting. We learned about the challenges and opportunities of developing a platform for online house rental. We also gained insights into the professional world and the expectations of our future employers and clients. We successfully completed our project and delivered a functional and secure platform that meets the needs and preferences of the users. We are proud of our achievements and we look forward to improving our platform and expanding our skills in the future.

BIBLIOGRAPHY

* Mrs. Tchinga Alice UML course AICS-Cameroon 2022-2023
* Mrs. EBENYE Vanessa Scientific Writing AICS-Cameroon 2022-2023
* OMG, OMG Unified Modelling Language 2.5, Object Management Group, March

2015, 794 pages.

* UML 2 For Dummies 2003.
* Internship report realized by ASANE DERICK ENOW Ba2, FINGERPRINT-ASSISTED ONLINE STUDENT FOLLOW-UP PLATFORM: Case study AICS-Cameroon in the Academic year 2022-2023.
* Internship report realized by NGATCHEUSSI TEUKAM ARISTOTE DANIEL PROGRESS Ba2, COMPUTERIZED MANAGEMENT OF AN ONLINE MEDICAL CONSULTATION PLATFORM in the Academic year 2022-2023.
* Mr. MESSIO MySQL course AICS-Cameroon 2022-2023.
* Mr. Agbor Computer Security AICS-Cameroon 2022-2023.

WEBOGRAPHY

* <https://htmlcolorcodes.com/> html color code for the colors use in my report
* <https://www.udemy.com/course/the-complete-python-django-rest-api-development-bootcamp/learn/lecture/24481164?start=165#overview> for learning the basics in Django
* <https://www.hippotechgroup.com/2021/01/how-to-get-your-cameroon-national-id-card-number.html> know the steps to have a card in Cameroon, September 28 2023
* <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/> know how to draw UML diagrams, September 27 2023
* <https://v2.crowdfind.com/site_login> study of related software doing the same application, September 25 2023
* <https://www.postgresql.org/download/> downloading PostgreSQL, 22 september 2023
* <https://www.visual-paradigm.com/download/?platform=windows&arch=64bit> downloading visual paradigm for the UML diagrams, 21 september 2023
* <https://github.com/chrispirit/LoFiCardApp/tree/donfack/assets/image> putting work project on git hub, September 13 2023
* <https://dribbble.com/get-started#welcomeDesigner> inspiration for the home page, 12 september 2023
* <https://reactnative.dev/docs/environment-setup> setting up react native, sep 6 2023
* <https://docs.expo.dev/get-started/installation/> documenting on expo sep 6 2023
* <https://staruml.io/download/> downloading star UML for the August 29 2023
* <https://www.javatpoint.com/uml-state-machine-diagram> learn state machine diagram, August 2023

TABLE OF CONTENT

[PART I 2](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147493559)

[INSERTION PHASE 2](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147493560)

[4](#_Toc147493561)

[INTRODUCTION 5](#_Toc147493562)

[3. WELCOMING AND INTEGRATION INTO ICES SARL 6](#_Toc147493563)

[4. GENERAL PRESENTATION OF ICES SARL 7](#_Toc147493564)

[4.1 History 7](#_Toc147493565)

[4.2 Mission 7](#_Toc147493566)

[4.3 Activities 7](#_Toc147493567)

[4.4 Services 8](#_Toc147493568)

[4.5 Subscription Service 9](#_Toc147493569)

[4.5.1 Microsoft Certification 9](#_Toc147493570)

[4.6 JURIDICAL STATUS 10](#_Toc147493571)

[4.7 GEOGRAPHICAL LOCATION 10](#_Toc147493572)

[5. PRESENTATION OF THE ADMINISTRATIVE AND FUNCTIONAL ORGANIZATION OF ICES\_SARL 12](#_Toc147493573)

[6. RESOURCES 14](#_Toc147493574)

[6.1 HARDWARE RESOURCES 14](#_Toc147493575)

[HARDWARE RESOURCES 14](#_Toc147493576)

[SOFTWARE RESOURCES 16](#_Toc147493577)

[SOFTWARE RESOURCES 16](#_Toc147493578)

[7. BRIEF PRESENTATION OF PROJECT 17](#_Toc147493579)

[CONCLUSION 18](#_Toc147493580)

[PART II 19](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147493581)

[EXISTING SYSTEM 19](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147493582)

[INTRODUCTION 22](#_Toc147493583)

[I. Brief Presentation of the project theme 23](#_Toc147493584)

[II. STUDY OF THE EXISTING SYSTEM 24](#_Toc147493585)

[III. CRITICISM OF THE EXISTING SYSTEM 25](#_Toc147493586)

[IV. PROBLEMATICS 25](#_Toc147493587)

[CONCLUSION 26](#_Toc147493588)

[PART III 27](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147493589)

[SPECIFICATION BOOK 27](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147493590)

[EXPRESSION OF NEEDS 31](#_Toc147493591)

[Functional needs 31](#_Toc147493592)

[NON-FUNCTIONAL NEEDS 31](#_Toc147493593)

[PART IV 41](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147493594)

[ANALYSIS PHASE 41](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147493595)

[INTRODUCTION 44](#_Toc147493596)

[2. PRESENTATION OF ANALYSIS METHODS 45](#_Toc147493597)

[2.1 OVERVIEW OF PROMINENT METHODOLOGIES 45](#_Toc147493598)

[I. UML 45](#_Toc147493599)

[II. MERISE 45](#_Toc147493600)

[III. Agile 45](#_Toc147493601)

[IV. SCRUM 45](#_Toc147493602)

[V. RAD (RAPID APPLICATION DEVELOPMENT) 45](#_Toc147493603)

[VI. XP (eXtreme Programming) 46](#_Toc147493604)

[VII. UP (Unified Process) 46](#_Toc147493605)

[2.2 INTRODUCING UML LANGUAGE 47](#_Toc147493606)

[2.2 SOFTWARE DEVELOPMENT PROCESS 50](#_Toc147493607)

[3. JUSTIFICATION OF ANALYSIS METHOD 53](#_Toc147493608)

[4. SYSTEM MODELLING 54](#_Toc147493609)

[4.1 FUNCTIONAL BRANCH 54](#_Toc147493610)

[1. Use Case Diagram 54](#_Toc147493611)

[a.) Definition 54](#_Toc147493612)

[b.) Formalism 54](#_Toc147493613)

[C. System’s Actors 57](#_Toc147493614)

[d.) General Use Case Diagram 58](#_Toc147493615)

[e.) Student Overview Use Case Diagram 59](#_Toc147493616)

[f.) Instructor Overview Use Case Diagram 60](#_Toc147493617)

[g.) Admin Overview Use Case Diagram. 61](#_Toc147493618)

[h.) Use Case Specification (Textual description) 62](#_Toc147493619)

[Formalism 62](#_Toc147493620)

[3.) Communication Diagram 66](#_Toc147493621)

[a.) Definition 66](#_Toc147493622)

[b.) Formalism 66](#_Toc147493623)

[c.) Components of a Communication Diagram. 67](#_Toc147493624)

[d.) System Login Communication diagram 68](#_Toc147493625)

[2.) Sequence Diagram 69](#_Toc147493626)

[a.) Definition 69](#_Toc147493627)

[b.) Formalism 69](#_Toc147493628)

[c.) Components of a Sequence Diagram. 70](#_Toc147493629)

[d.) Sequence diagram: Signup 72](#_Toc147493630)

[e.)Sequence diagram: Login 73](#_Toc147493631)

[2.) Activity Diagram 74](#_Toc147493632)

[a.) Definition 74](#_Toc147493633)

[b.) Formalism 74](#_Toc147493634)

[74](#_Toc147493635)

[c.) SignUp Activity Diagram 77](#_Toc147493636)

[c.) Login Activity Diagram 78](#_Toc147493637)

[CONCLUSION 79](#_Toc147493638)

[PART IV 80](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147493639)

[CONCEPTION PHASE 80](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147493640)

[INTRODUCTION 83](#_Toc147493641)

[2. TECHNICAL BRANCH 84](#_Toc147493642)

[2.1) CAPTURE OF TECHNICAL NEEDS 84](#_Toc147493643)

[a.) Physical Architecture 84](#_Toc147493644)

[b.) Logical architecture 85](#_Toc147493645)

[2.2 GENERIC DESIGN 86](#_Toc147493646)

[a.) Hardware diagram of the system 86](#_Toc147493647)

[b.) High Level Architectural Diagram of the Software 87](#_Toc147493648)

[3.) IMPLEMENTATION BRANCH 88](#_Toc147493649)

[3.1) PRELIMINARY DESIGN 88](#_Toc147493650)

[1.) State Machine Diagram 88](#_Toc147493651)

[a.) Definition 88](#_Toc147493652)

[b.) Formalism 88](#_Toc147493653)

[88](#_Toc147493654)

[c.) Components of a State Machine Diagram 89](#_Toc147493655)

[d.) System Login State Machine Diagram 89](#_Toc147493656)

[90](#_Toc147493657)

[d.) System SignUp State Machine Diagram 91](#_Toc147493658)

[91](#_Toc147493659)

[2.) Package Diagram 92](#_Toc147493660)

[a.) Definition 92](#_Toc147493661)

[b.) Formalism 92](#_Toc147493662)

[92](#_Toc147493663)

[c.) Components of a Package Diagram 93](#_Toc147493664)

[d.) Web Package Diagram 94](#_Toc147493665)

[3.2) DETAILED DESIGN 95](#_Toc147493666)

[1.) Class Diagram 95](#_Toc147493667)

[a.) Definition 95](#_Toc147493668)

[b.) Formalism 95](#_Toc147493669)

[c.) Components of a Class diagram 96](#_Toc147493670)

[d.) ICES ACADEMY Class diagram 98](#_Toc147493671)

[98](#_Toc147493672)

[CONCLUSION 99](#_Toc147493673)

[PART V 100](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147493674)

[Preface 101](#_Toc147493675)

[INTRODUCTION 103](#_Toc147493676)

[2.) DEPLOYMENT DIAGRAM 104](#_Toc147493677)

[a.) Definition 104](#_Toc147493678)

[b.) Formalism 104](#_Toc147493679)

[104](#_Toc147493680)

[c.) Components of a Deployment Diagram 105](#_Toc147493681)

[d.) System Deployment diagram 106](#_Toc147493682)

[106](#_Toc147493683)

[3.) COMPONENT DIAGRAM 107](#_Toc147493684)

[a.) Definition 107](#_Toc147493685)

[b.) Formalism 107](#_Toc147493686)

[107](#_Toc147493687)

[c.) Components of a Deployment Diagram 108](#_Toc147493688)

[d.) System Deployment diagram 109](#_Toc147493689)

[109](#_Toc147493690)

[4.) PRESENTATION OF THE TOOL USED 110](#_Toc147493691)

[a.) SOFTWARE RESOURCES 110](#_Toc147493692)

[a.) HARDWARE RESOURCES 111](#_Toc147493693)

[5.) PRESENTATION OF THE LANGUAGE USED 111](#_Toc147493694)

[CONCLUSION 112](#_Toc147493695)

[PART VI 113](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147493696)

[FUNCTIONALITY TEST 113](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147493697)

[Preface 114](#_Toc147493698)

[INTRODUCTION 116](#_Toc147493699)

[2.) PRESENTATION OF TESTING METHOD 117](#_Toc147493700)

[3.) SYSTEM TESTING 118](#_Toc147493701)

[a.) DB Testing 118](#_Toc147493702)

[118](#_Toc147493703)

[PART VII 119](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147493704)

[USER GUIDE 119](file:///C:\Users\acer\Desktop\REPORT\MAIN%20FOOTER--HEADER.docx#_Toc147493705)

[INTRODUCTION 122](#_Toc147493706)

[I.) INSTALLATION GUIDE 123](#_Toc147493707)

[a.) XAMPP SERVER INSTALLATION 123](#_Toc147493708)

[USER GUIDE 128](#_Toc147493709)

[Launching the Application 128](#_Toc147493710)

[HOME PAGE VIEW 129](#_Toc147493711)

[ADMIN PAGE VIEW 129](#_Toc147493712)

[CONCLUSION 130](#_Toc147493713)