

University of Hassiba Ben Bouali Chlef
Faculty Of Science And technology
Department of Computer science

Internships Management System

Mohammed ELhaouari

Submitted in part fulfilment of the requirements for the degree of
Licence in Computing of the University of Hassiba Ben Bouali Chlef and
May 2016

Abstract

We live in a jet age when results are expected to turn in speedily as evident in the various fields of study such as engineering, accounting, architecture , medicine to mention a few, where this concept of Computers has made tremendous impact. It is therefore imperative to embrace and appreciate it for the fact that it has made the work of humans a lot easier, faster with a greater degree of accuracy and precision. Computer Technology had also an enormous impact on Education in the 21 century, and this concept of computers is life made easy in itself exacting a kind of intelligence that transcends human ability naturally. With the availability of computer system, most institution has access to internet facilities which is also a necessity, Universities though still uses papers and manual process for managing internships and student projects adapting the traditional way. Manual process means, the student or intern's candidate need to fill a paper form, and also need to submit it by hand to the supervisor they have chosen. During the end of study project, intern's need to keep their supervisor updated about their daily routine of the project and the work progress by writing it and paste any attachment. Finally, the final work will be examined by university supervisor and a jury of professors and professionals. Our Internship Management System is a management system built to automate the process of collecting, reviewing and managing applications for internships as well as managing the ongoing lifecycle of the project.

Acknowledgements

I would like to thank my Lord for giving me the opportunity and skill with which to write this Report, and Letting me see it to completion. I would like to thank my mother for countless cups of coffee, good council and lots of patience, and for her Encouragement and prayer. I would like to thank my father for the hope and love, and for his encouragement and prayer. I would like to thank my friends and mentors, especially Dr Mohammed Aridj. I would Like to thank my jury professors Dr Danouni Nassim and Dr Slimani Ahmed Finally,I would Like to thank everyone who has helped me despite my failure to mention them here,

‘A wise man will make more opportunities than he finds.’

Francis Bacon

Contents

| | |
|--|----------|
| Abstract | i |
| Acknowledgements | i |
| Overview | 1 |
| Introduction | 1 |
| Problem Statement | 2 |
| Objectives | 3 |
| Scopes | 3 |
| Thesis Organisation | 3 |
| 1 Introduction | 5 |
| 1.1 Internships | 5 |
| 1.1.1 Definition | 5 |
| 1.1.2 Importance of Internships | 6 |
| 1.2 Scope Organization(UHBC) | 6 |
| 1.3 Current Internship Management System Review: | 7 |
| 2 Methodology | 9 |
| 2.1 Modelling And Model-driven Architecture | 9 |

| | | |
|---------|---|----|
| 2.2 | Software Development Approach Considered | 9 |
| 2.2.1 | Object Oriented Approach | 9 |
| 2.2.2 | Structured Approach | 10 |
| 2.2.3 | Rationale For Adopting The Object Oriented Approach | 11 |
| 2.3 | Modelling Languages Considered | 12 |
| 2.3.1 | Unified Modelling Language | 12 |
| 2.3.2 | Rationale For Choosing UML | 14 |
| 2.3.3 | Use Case Diagram | 14 |
| 2.3.3.1 | Definition | 14 |
| 2.3.3.2 | Actors | 14 |
| 2.3.3.3 | Our Diagram | 15 |
| 2.3.4 | Class Diagrams | 16 |
| 2.3.4.1 | Definition | 16 |
| 2.3.4.2 | Classes | 16 |
| 2.3.4.3 | Our Class Diagram | 17 |
| 2.3.5 | Activity Diagrams | 18 |
| 2.3.5.1 | Definition | 18 |
| 2.3.5.2 | An Activity | 18 |
| 2.3.5.3 | Authentication Diagram | 19 |
| 2.3.5.4 | Student Register Activity Diagram | 20 |
| 2.3.5.5 | Student Enrol To Theme Activity Diagram | 21 |
| 2.3.5.6 | Supervisor Post Theme Activity Diagram | 22 |
| 2.3.5.7 | Supervisor Endorse Theme Activity Diagram | 23 |
| 2.3.5.8 | Reviewer Validate Theme Activity Diagram | 24 |

| | | |
|----------|---|-----------|
| 2.3.6 | Sequence Diagram | 25 |
| 2.3.6.1 | Definition | 25 |
| 2.3.6.2 | Authentication Sequence Diagram | 26 |
| 2.3.6.3 | Student Register Sequence Diagram | 27 |
| 2.3.6.4 | Supervisor Post Theme Sequence Diagram | 28 |
| 2.3.6.5 | Supervisor Endorse Theme Sequence Diagram | 29 |
| 2.3.6.6 | Reviewer Validate Theme Sequence Diagram | 30 |
| 2.4 | Data Modelling | 31 |
| 2.4.1 | Entity Relationship Diagram (ERD) | 31 |
| 2.4.1.1 | An Entity | 31 |
| 2.4.1.2 | Relationships | 31 |
| 2.4.1.3 | Attributes | 31 |
| 2.4.2 | The Cardinality | 31 |
| 2.4.3 | IMS ER Diagram | 32 |
| 2.5 | Architecture Of Implementation | 33 |
| 2.5.1 | Model, View And Controller | 33 |
| 2.5.1.1 | Model | 33 |
| 2.5.1.2 | View | 33 |
| 2.5.1.3 | Controller | 33 |
| 2.5.2 | Rational For Choosing MVC | 34 |
| 2.5.3 | Conclusion | 35 |
| 3 | Implementation | 36 |
| 3.1 | Programming Languages and Tools Used | 36 |
| 3.1.1 | Programming Languages | 36 |

| | | |
|---------|---------------------------------|----|
| 3.1.1.1 | HTML | 36 |
| 3.1.1.2 | CSS | 36 |
| 3.1.1.3 | SASS | 37 |
| 3.1.1.4 | Javascript | 37 |
| 3.1.1.5 | Ajax | 37 |
| 3.1.1.6 | PHP | 37 |
| 3.1.1.7 | MYSQL | 38 |
| 3.1.2 | Tools | 38 |
| 3.1.2.1 | XAMPP | 38 |
| 3.1.2.2 | Git | 38 |
| 3.1.2.3 | Astah Professional | 38 |
| 3.1.2.4 | Sublime Text 3 | 39 |
| 3.1.2.5 | Google Chrome browser | 39 |

List of Tables

| | | |
|-----|--|----|
| 2.1 | Object Oriented Approach Advantages And Disadvantages | 10 |
| 2.2 | Structured (Traditional) Approach Advantages And Disadvantages | 10 |
| 2.3 | The Unified Modeling Language Diagrams | 12 |
| 2.4 | UML Advantages And Disadvantages | 13 |
| 2.5 | Internship Management System Actors | 14 |
| 2.6 | Advantages And Disadvantages Of MVC Architectural Pattern | 34 |

List of Figures

| | | |
|------|--|----|
| 2.1 | The Unified Modeling Language Diagrams | 13 |
| 2.2 | UML Use Case Diagram | 15 |
| 2.3 | Internship Management System Class Diagram | 17 |
| 2.4 | Internship Management System Authentication Activity Diagram | 19 |
| 2.5 | Student Register Activity Diagram | 20 |
| 2.6 | Student Enrol To Theme Activity Diagram | 21 |
| 2.7 | Supervisor Post Theme Activity Diagram | 22 |
| 2.8 | Supervisor Endorse Theme Activity Diagram | 23 |
| 2.9 | Reviewer Validate Theme Activity Diagram | 24 |
| 2.10 | Authentication Sequence Diagram | 26 |
| 2.11 | Student Register Sequence Diagram | 27 |
| 2.12 | Supervisor Post theme Sequence Diagram | 28 |
| 2.13 | Supervisor Endorse Theme Sequence Diagram | 29 |
| 2.14 | Reviewer Validate Theme Sequence Diagram | 30 |
| 2.15 | IMS ER Diagram | 32 |
| 2.16 | MVC Structure Diagram | 34 |

Overview

Introduction

Internships are the way to explore or expand the related knowledge and skills required in a real world situation and experience to enter into a particular career field. It is basically for undergraduate student, or a trainee, this not only for gaining experience, but also expose them to fine if they totally interest in the field. Regarding to UHBC University Computer Science Department, the number of students will be out in the industry is roughly 100 people. So, every year about 100 students will be on interns, but this is depending on how many people qualify to register it. All students must pass the entire Modules they have taken before they go for interns. Internships represent three years of theoretical and practical training. Their objective is to help students develop analytical, research, as well as organizational and presentation skills applied to the corporate world. Students dedicate more than 180 hours (a whole semester) to their consulting project, thus building a capacity to conduct personal research on a specific domain. Then, they present their results in at most a 50-page document and undergo an interview before a jury of professors and professionals.

The topics are related to one of The Computer science fields of specialization:

1. Artificial Intelligence (AI)
2. Computer Architecture and Engineering (ARC)
3. Database Management Systems (DBMS)
4. Graphics (GR)
5. Human-Computer Interaction (HCI)
6. Operating Systems And Networking (OSNT)

7. Programming Systems (PS)

8. Scientific Computation (SCI)

During the end of the study project, intern's need to keep their supervisor updated about their daily routine of the project and the work progress by writing it and paste any attachment. Finally, the work will be examined by the university supervisor and a jury of professors and professionals. Manual process needs to be done in this system, manual process means, the student or intern's candidate need to fill a paper form, and also need to submit it by hand to the supervisor they have chosen. Applying a manual process may cause some disadvantages, because it required the user to write it in a paper, instead of write it in the computer. Error in filling in the form can be avoided if it is filled with computer. And the data cannot be directly used, but need to be rewritten before can be used, this way will have a lot of work on it. Other than that, the problem of losing or misplace the form can be avoided, or damaging the hard copy form will not occur.

Problem Statement

In today's world where information and communication technology systems have become ubiquitous with different vendors providing various kinds of management software with different range of applicability, it is rather surprising that internship at UHBC University is managed without one. It is easy to find a staff typing each applicant's internship offer letter in Microsoft word. Given the vast number of applicant's that are accepted per year, this approach is laborious. Furthermore, the usage of Microsoft Excel as a database for storing records further makes the whole process very difficult to manage. In addition, the manual creation of various reports not only makes the process time consuming, but also is wastage of human resource effort that would have been used to perform other tasks. When applying to the internship, students need to complete the registration process. This process needs to be done manually, which mean students need to fill and complete it by write it in a document. After receiving an accordance letter from the company, students need to inform the faculty internship coordinator for approving. While a student undergoing the training, they need to document what they are doing every day in a report. Other than that, student performance also had to be marked by the university supervisor every week. The mark will be given in the final work.

Objectives

The impact of using computer applications to the teaching and learning processes is becoming important issues and this project attempts to provide a solution to these problems. Thus we aim to develop a prototype of an online internship management system that works by allowing the university to deploy a web portal, to collect applicants data, review that data, make a decision, and then continue to interact with applicants and reviewers. The goal of the system is to enable online interaction with applicants and their supervisors and automate the entire process; thereby eliminating a huge amount of paperwork and manual tasks.

Scopes

The scope of the system is to provide a university with a friendly set of web pages that are easy to navigate and at the same time provides sufficient depth and information about the system and how it works. The scopes of the project are:

- Department of Computer Science - Faculty of Science - UHBC University
- Web base application
- Users:
 - UHBC's Students.
 - Faculty Supervisors.
 - Internship Coordinators.
 - System administrators.

Thesis Organisation

This thesis consists of 3 chapters ranging from Chapter 1 to Chapter 3. Chapter 1 gives an overview of the study conducted and review for the present system used to manage internships and its weakness and problems. And a description of the different methods available to solve the problem, their merits and demerits. Meanwhile, Chapter 2 reveals the techniques and the use cases of our system, and the

combination of the methods by which the project was conducted and controlled, and also, the methods in software development that was applied. Chapter 3 will provide a description of how the system was implemented.

Chapter 1

Introduction

1.1 Internships

1.1.1 Definition

The thesis is the implementation of a structured and rigorous research project in a particular field, allowing the students to improve their knowledge in their field of study. For many students, an internship is a first opportunity to get to know an industry, a job role and the working culture of a company, whilst making contacts and applying skills developed during university. It is important to consider internships and make applications to companies you both want to work at and believe will provide you with a good platform for getting a graduate job upon graduation. The master's thesis must demonstrate that the candidate has the aptitude for research and knows well write and present the results of his work. The ability of the candidate looking as demonstrated by the memory:

1. The ability to conduct systematic reviews.
2. The contribution to the advancement of knowledge.
3. Quality of writing language.
4. The quality of the material and typographical arrangements.
5. 5 The overall quality of the work (title, abstract, etc.).

1.1.2 Importance of Internships

Graduation internships are very important, because they substantially increase the chance of graduates finding employment. Internships are also very useful to interns themselves as they offer the chance to find out what working for a particular company, or within a certain industry, is really like. Internships also allow interns to make contacts with managers and recruiters, which can later be used to negotiate full-time employment. Internships are a great way for students to test their skills in real-life situations, explore career options and gain an insight into a company or organization. Taking an internship can also significantly increase your chances of getting a full-time graduate job. It is not at all unusual for employers make full-time job offers to a large proportion of their interns.

1.2 Scope Organization(UHBC)

Passing from the status of an Institute at (1983), to a University Centre (1988) status, to a University of type B (2001) and finally to University of type A in (2011), the University of Chlef was able to respond and adapt to regional needs and expectations of the public power.

The number, which was 200 students in 1983, reached more than 24,000 students today, including 2500 subscribed to Masters and 600 in Post- graduation across seven faculties and two institutes supervised by more than 784 teachers.

- A comprehensive university
 - More than 31 courses of study.
 - Robust programs and recognized exchanges.
 - Customized training programs.
 - More than 136,000 library books.
 - 08 databases
- A rich and diverse community:
 - More than 24 000 students
 - Over 25,000 training places.
 - More than 784 teachers.

- A unique research environment:
 - More than 152 researchers.
 - 37 research projects.
 - 16 research laboratories.
 - International conventions.
- An exceptional place:
 - More than 10,618 students spread over 06 university dorms.
 - A fleet of 107 buses to 10,097 subscribers.
 - 07 university restaurants with a capacity of 3980 seats.

1.3 Current Internship Management System Review:

This project was initiated mainly to address the problems encountered in the management of Internship at Hassiba Ben Bouali University of Chlef. Below are some of the problems that can be identified with the current system and the ways that this project attempts to solve: The manual creation of each applicant's internship offer letter using Microsoft Word is a real waste of time and human resource effort. The new system will offer the possibility to generate this letter automatically. There is a lack of data independence and efficient access to the current system. Records stored in Microsoft Spreadsheet in windows may not open properly in OpenOffice in Linux. The new system will run on any platform efficiently. The fact that an applicant cannot apply and track his application online makes the whole process too demanding for applicants. Collecting paper documents from applicants can easily lead to the loss of applicant's documents considering the vast number of applicants who apply regularly. With the new system, applicants will be able to apply and check the status of their application online. Changes made to data cannot easily be seen by other staffs in the same unit and as far as security is concerned, there is the tendency that people without the right grant or permission can easily gain access to the stored data if they manage to gain access to the management staff computer. The new system will employ a unified database, making it easy for changes to be seen by everyone. Also, security will be enforced by granting each user a username and password. The new system will offer the possibility to track and trace human errors. Also, because many aspects will be automated human error is reduced. The fact that the current system is not web-based, it is impossible to manage an

internship remotely. Since the new system is web-based and will run on a web server, remote access is possible via internet connection. The problem of data redundancy is also addressed as records will be stored in a well normalized database. Thus, the new system benefits from the advantages of a database management system.

Chapter 2

Methodology

2.1 Modelling And Model-driven Architecture

Software systems are reaching such a high degree of complexity that even current third generation programming languages, like Java or C, are becoming unable to efficiently support the creation of such systems. One of the problems with such current languages is that they are still too oriented toward specifying how the solution should work, instead of what the solution should be. This leads to a need for mechanisms and techniques that allow the developer to abstract over the programming language itself and focus on creating a simple solution to a certain problem.

2.2 Software Development Approach Considered

In order to analyse, design and implement the software system, two approaches to software development were considered: Object Oriented Approach and Structured Approach.

2.2.1 Object Oriented Approach

The object-oriented approach is a development strategy based on the concept that systems should be built from a collection of reusable parts called objects (AMBLER, 2004).

| Advantages | Disadvantages |
|---|--|
| Enables development of complex systems. | The significant skill set is required. |
| Wide industry acceptance. | No single language dominates the landscape (Although Java, C#, C++, PHP are clearly popular and here to stay). |
| Mature, proven technology. | Not all IT professionals, in particular some within the data community, accept it. |
| A wide range of development languages and tools to choose from. | Technical "impedance mismatch" with Structured technologies and RDBs. |

Table 2.1: Object Oriented Approach Advantages And Disadvantages

2.2.2 Structured Approach

The structured approach is a development strategy based on the concept that a system should be separated into two parts: data (modeled using a data model) and functionality (modeled using a process model). Following the structured approach, you develop applications in which data are separate from behavior, both in the design model and in the system implementation (AMBLER, 2004). the following table shows Structured (Traditional) Approach Advantages And Disadvantages

| Advantages | Disadvantages |
|--------------------------------|--|
| Reduces complexity. | Unsuitable for very complex system. |
| Focuses on the idea. | Less dominance in the market.Object orientation is more dominant in todays market. |
| Relies less on artistry. | Assume development is a sequential process. |
| Provides a future orientation. | |

Table 2.2: Structured (Traditional) Approach Advantages And Disadvantages

2.2.3 Rationale For Adopting The Object Oriented Approach

The Object Oriented Approach was adopted over the Structured Approach because of the following reasons:

- The complexity of this project.
- My familiarity with the approach.
- Much more suitable for this project.
- The dominance of object oriented paradigm and its related technologies in market and industry.
- The technical and theoretical capabilities of the method.
- The wide acceptance in industry.
- Wide range of development languages and tools to choose from.

2.3 Modelling Languages Considered

In order to model the system, we have chosen the de-facto modelling language: UML

2.3.1 Unified Modelling Language

The Unified Modeling Language (UML) Superstructure (typically called only UML) is a general-purpose modeling language. Originally designed to specify, visualize, and document software systems, it is also used as a basis for a variety of tools, methodologies, and research efforts regarding software development and source code generation. UML 2.5 (the current version as of the writing of this dissertation) is composed of 14 different kinds of diagram (as illustrated in Figure 2.1), divided into two categories:

- Structure diagrams, which deal with the static structure of the objects in the system being modeled.
- Behavior diagrams, which specify the dynamic aspects and behavior of the system's objects.

| Structure diagrams | Behavior diagrams |
|------------------------------|-------------------------------|
| Class diagram. | Activity diagram. |
| Component diagram. | Communication diagram, |
| Composite structure diagram. | Interaction overview diagram. |
| Deployment diagram. | Sequence diagram. |
| Object diagram. | State machine diagram. |
| Package diagram. | Timing diagram. |
| Profile diagram. | Use case diagram. |

Table 2.3: The Unified Modeling Language Diagrams

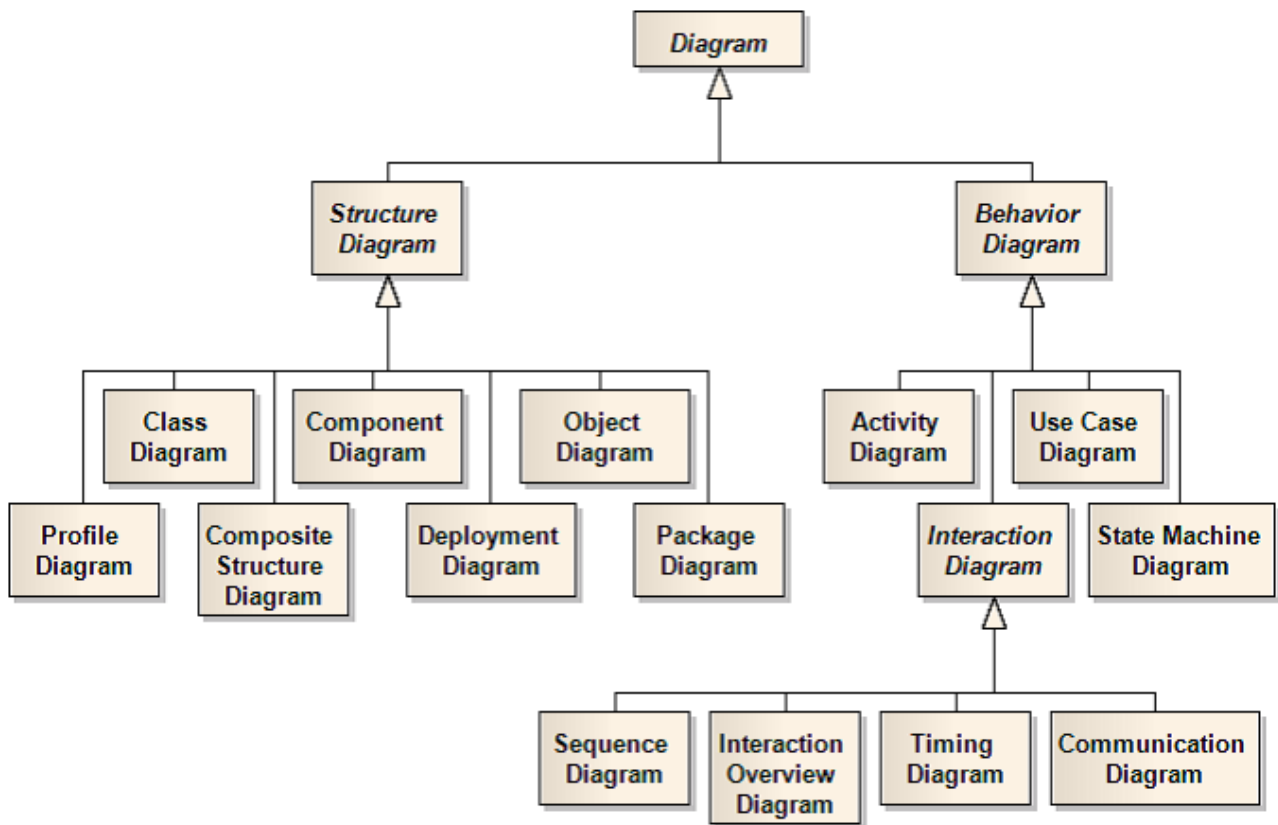


Figure 2.1: The Unified Modeling Language Diagrams

| Advantages | Disadvantages |
|--|---|
| UML breaks complex system into discrete pieces that can be understood easily. Handover of the system to new team becomes easier. | UML is an Object Modeling language which covers only a specific section in system development. |
| Complex systems can be understood by disparate developers who are working on different platforms. | UML does not define a standard file format, meaning that each UML tool vendor stores the representation of its UML model in a proprietary format. |
| UML model is not system or platform specific. It unifies all disparate developers under one roof. | Using multiple models/diagrams make it difficult to keep them consistent with each other and the code and much code have to be added by hand. |

Table 2.4: UML Advantages And Disadvantages

2.3.2 Rationale For Choosing UML

The Unified Modeling Language was used to model the system. Reasons for choosing UML over Merise are explained below.

- My Familiarity with UML.
- The dominance of UML in the market. Merise is only well known in France.
- Stakeholder specification.

2.3.3 Use Case Diagram

2.3.3.1 Definition

Use cases are a way to capture system functionality and requirements in UML. Use case diagrams consist of the named pieces of functionality (use cases), the persons or things invoking the functionality (actors), and possibly the elements responsible for implementing the use cases (subjects).

2.3.3.2 Actors

A use case must be initiated by someone or something outside the scope of the use case. This interested party is called an actor. An actor doesn't need to be a human user; any external system or element outside of the use case may trigger the use case (or be the recipient of use case results) and should be modeled as an actor. In our internship management system we have the following actors

| Actor | Definition |
|---------------|--|
| Student | Students of UHBC university that are applying for interns |
| Supervisor | Professors who supervise students during their internship |
| Reviewer | The head of the scientific consul who review proposed subjects |
| Administrator | University employee who administer the system |

Table 2.5: Internship Management System Actors

2.3.3.3 Our Diagram

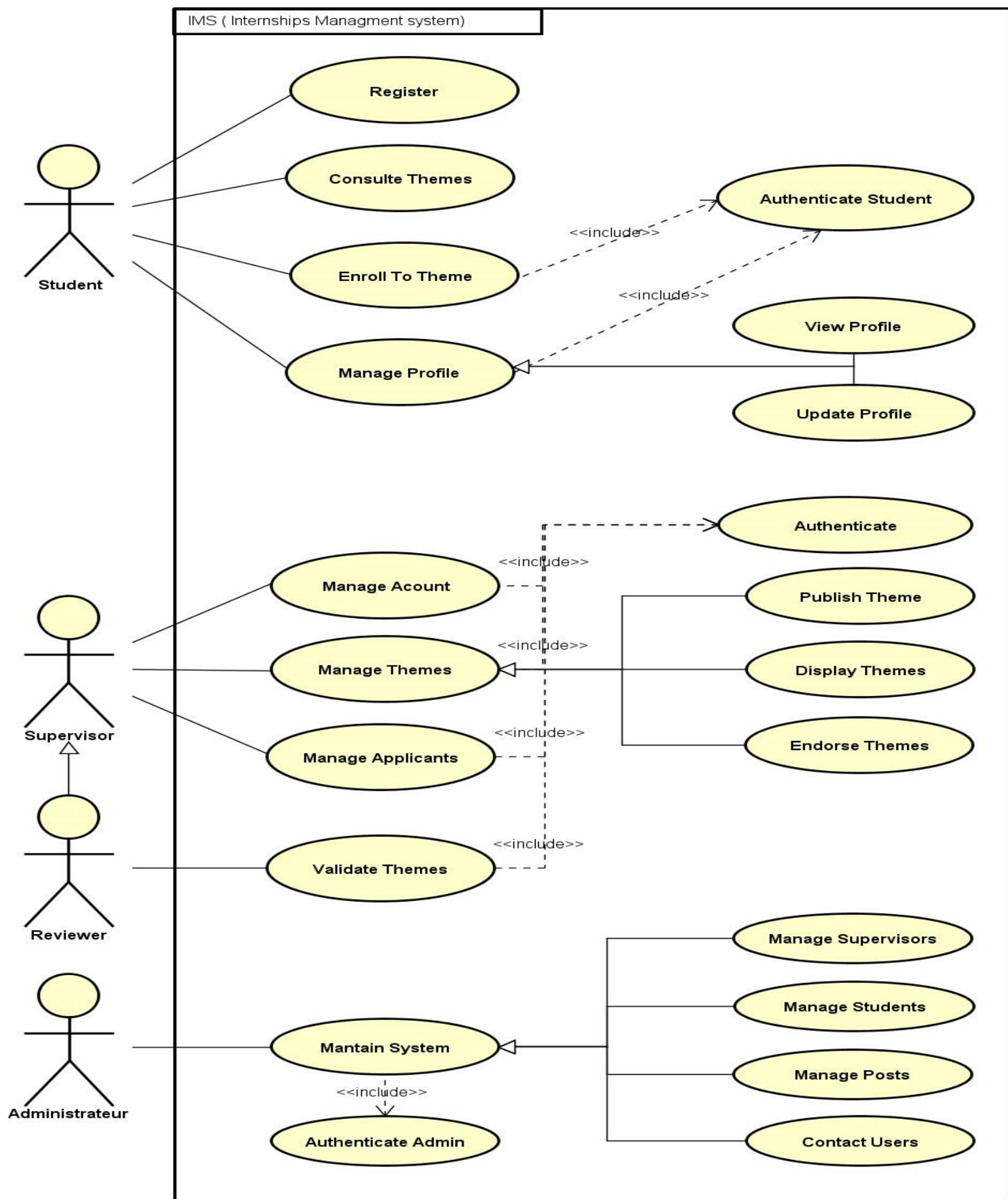


Figure 2.2: UML Use Case Diagram

2.3.4 Class Diagrams

2.3.4.1 Definition

Class diagrams are one of the most fundamental diagram types in UML. Class diagrams are used in both the analysis and the design phases. During the analysis phase, a very high-level conceptual design is created. At this time, a class diagram might be created with only the class names shown or possibly some pseudo code-like phrases may be added to describe the responsibilities of the class. The class diagram created during the analysis phase is used to describe the classes and relationships in the problem domain, but it does not suggest how the system is implemented. By the end of the design phase, class diagrams that describe how the system to be implemented should be developed. The class diagram created after the design phase has detailed implementation information, including the class names, the methods and attributes of the classes, and the relationships among classes.

2.3.4.2 Classes

A class represents a group of things that have common state and behavior. We can think of a class as a blueprint for an object in an object-oriented system. In UML speak, a class is a kind of classifier. For example, Volkswagen, Toyota, and Ford are all cars, so we can represent them using a class named Car. Each specific type of car is an instance of that class, or an object. A class may represent a tangible and concrete concept, such as an invoice; it may be abstract, such as a document or a vehicle.

2.3.4.3 Our Class Diagram

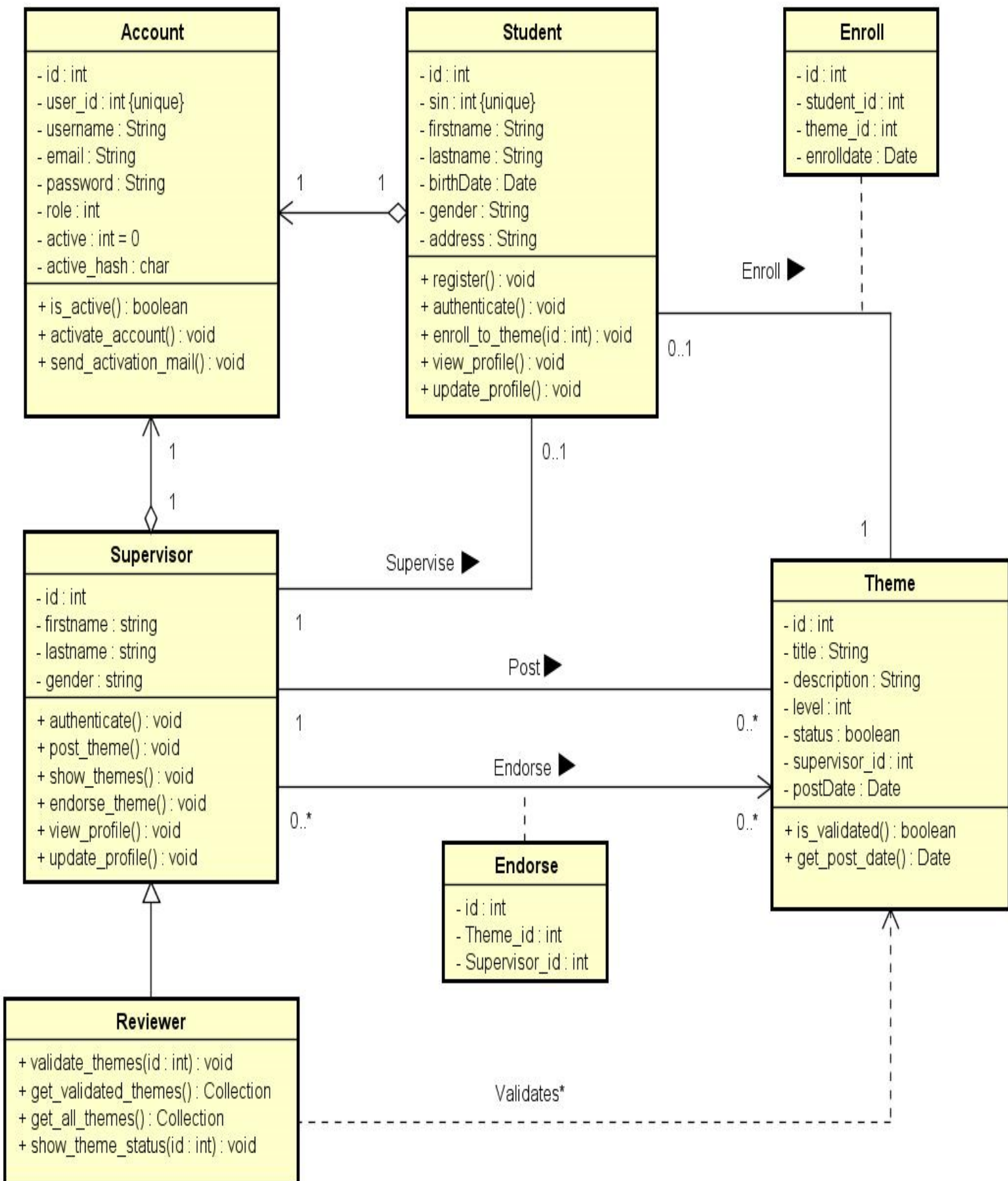


Figure 2.3: Internship Management System Class Diagram

2.3.5 Activity Diagrams

2.3.5.1 Definition

Activity diagrams are used during the design phase of complex methods. Alternately, the activity diagram can also be used during analysis to break down the complex flow of a use case. Through an activity diagram, the designer/analyst specifies the essential sequencing rules the method or use case has to follow.

2.3.5.2 An Activity

An activity is a single step that needs to be done, whether by a human or a computer (Fowler, 2000). Incoming transitions (an incoming arrow) trigger the activity. If there are several incoming transitions, any of these can trigger the activity independent of the others. (Oestereich, 2001)

2.3.5.3 Authentication Diagram

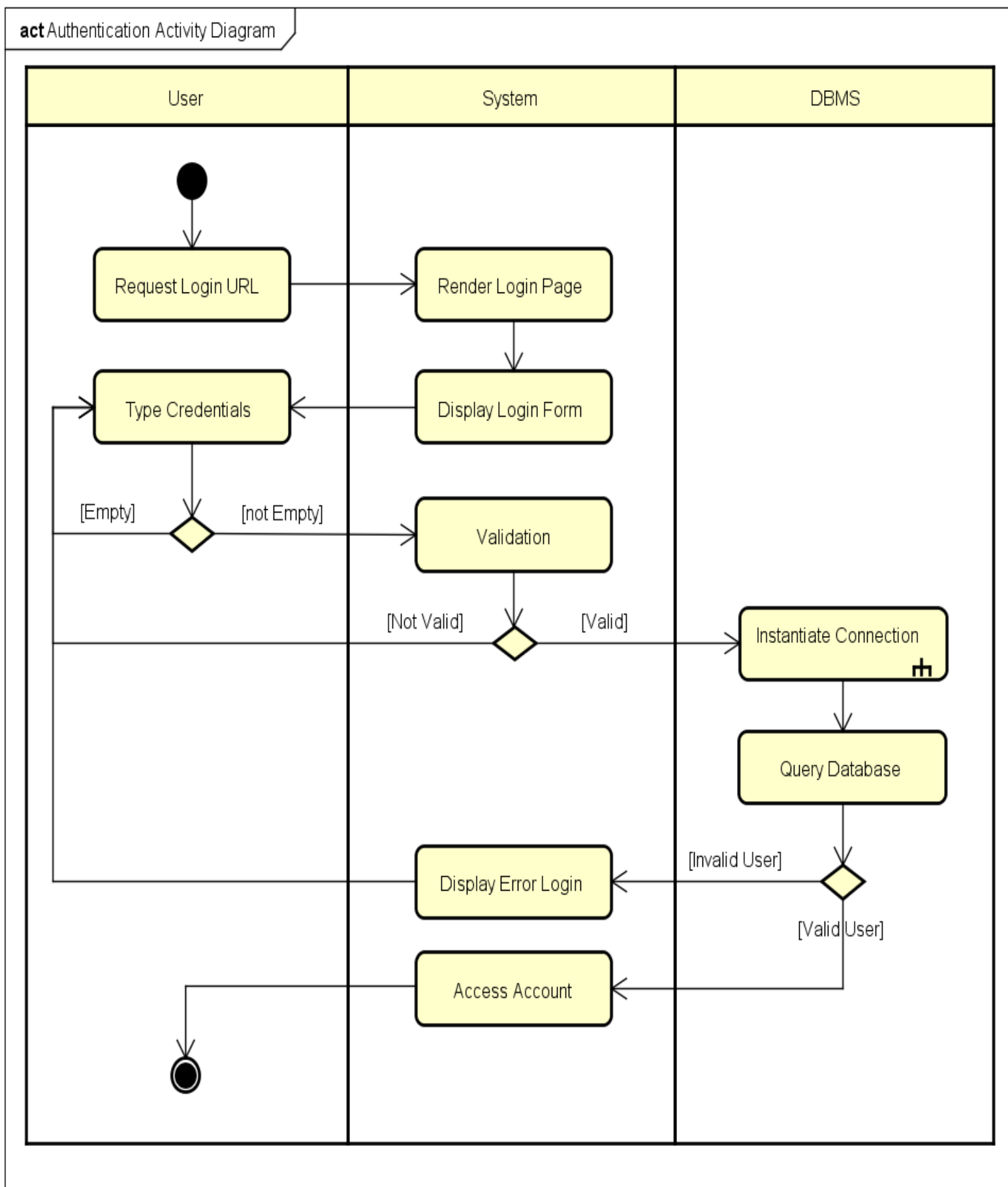


Figure 2.4: Internship Management System Authentication Activity Diagram

2.3.5.4 Student Register Activity Diagram

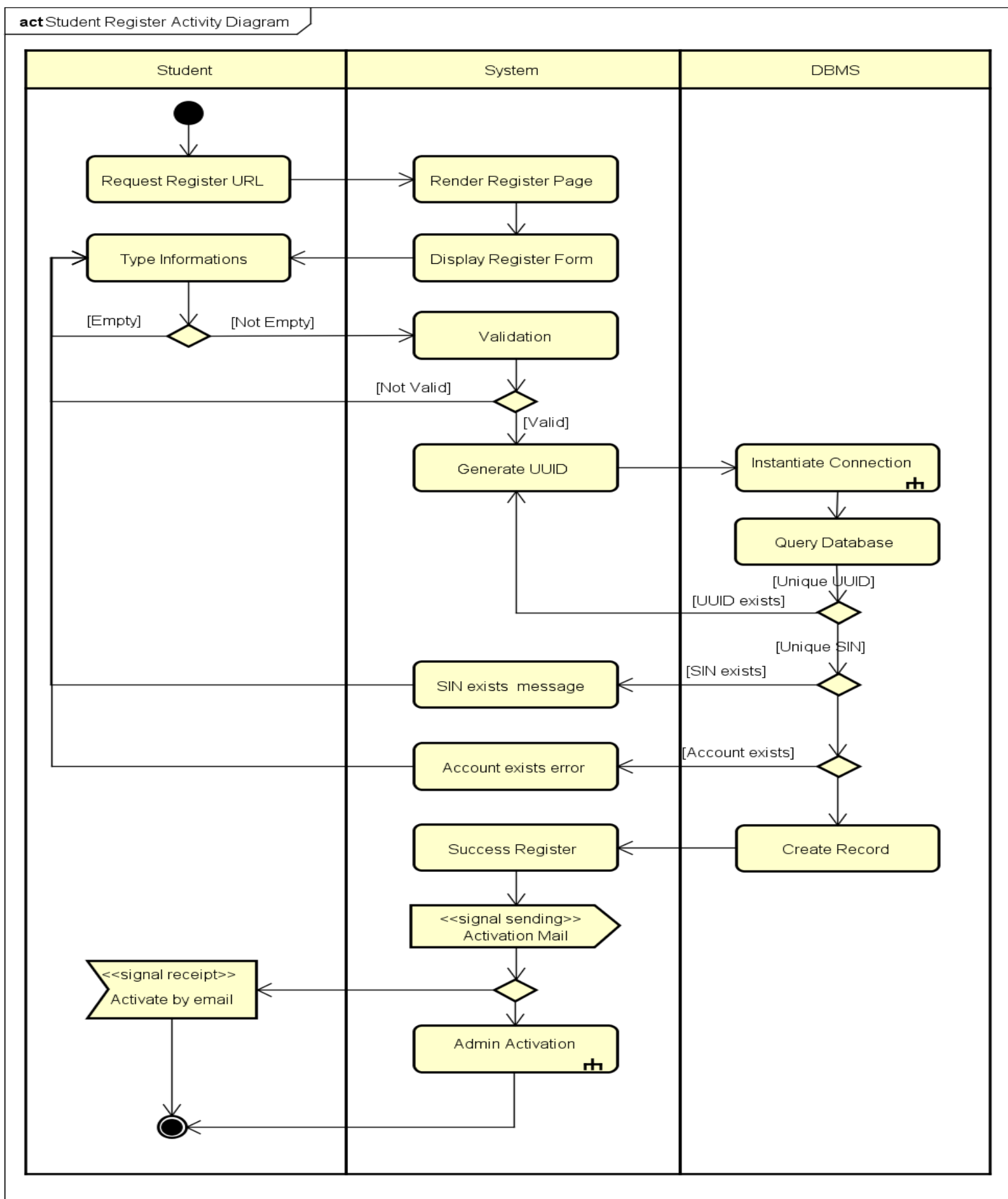


Figure 2.5: Student Register Activity Diagram

2.3.5.5 Student Enrol To Theme Activity Diagram

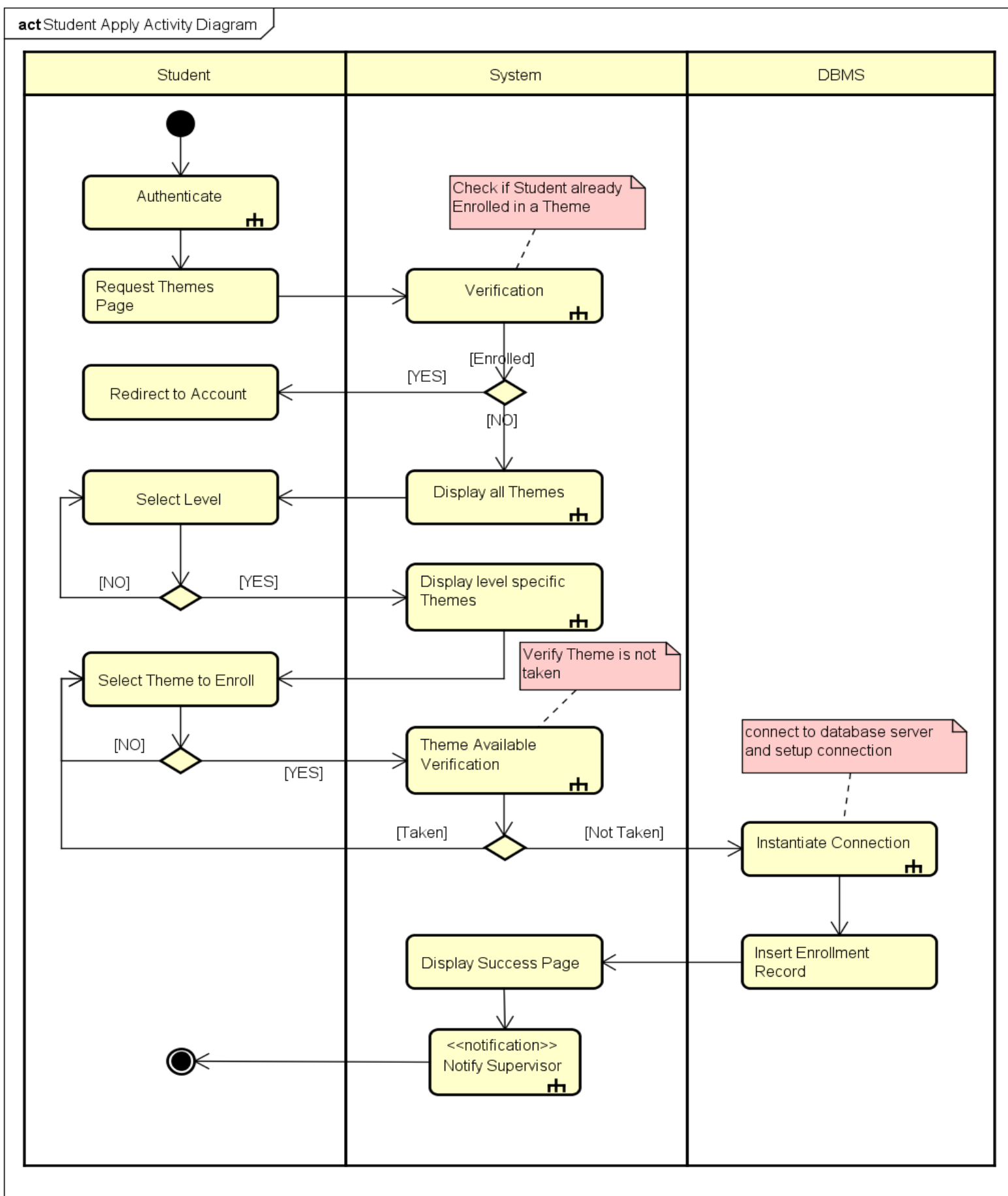


Figure 2.6: Student Enrol To Theme Activity Diagram

2.3.5.6 Supervisor Post Theme Activity Diagram

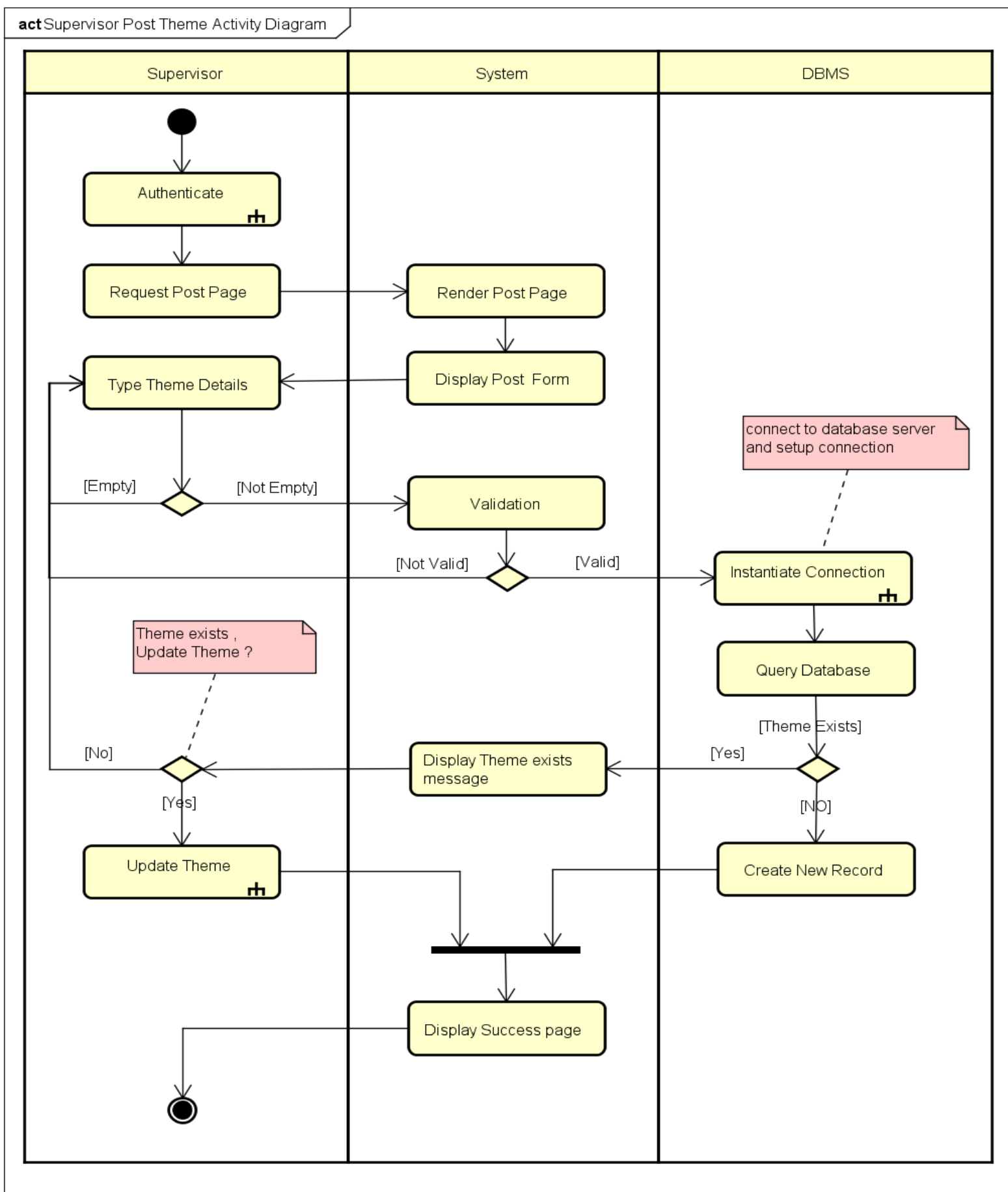


Figure 2.7: Supervisor Post Theme Activity Diagram

2.3.5.7 Supervisor Endorse Theme Activity Diagram

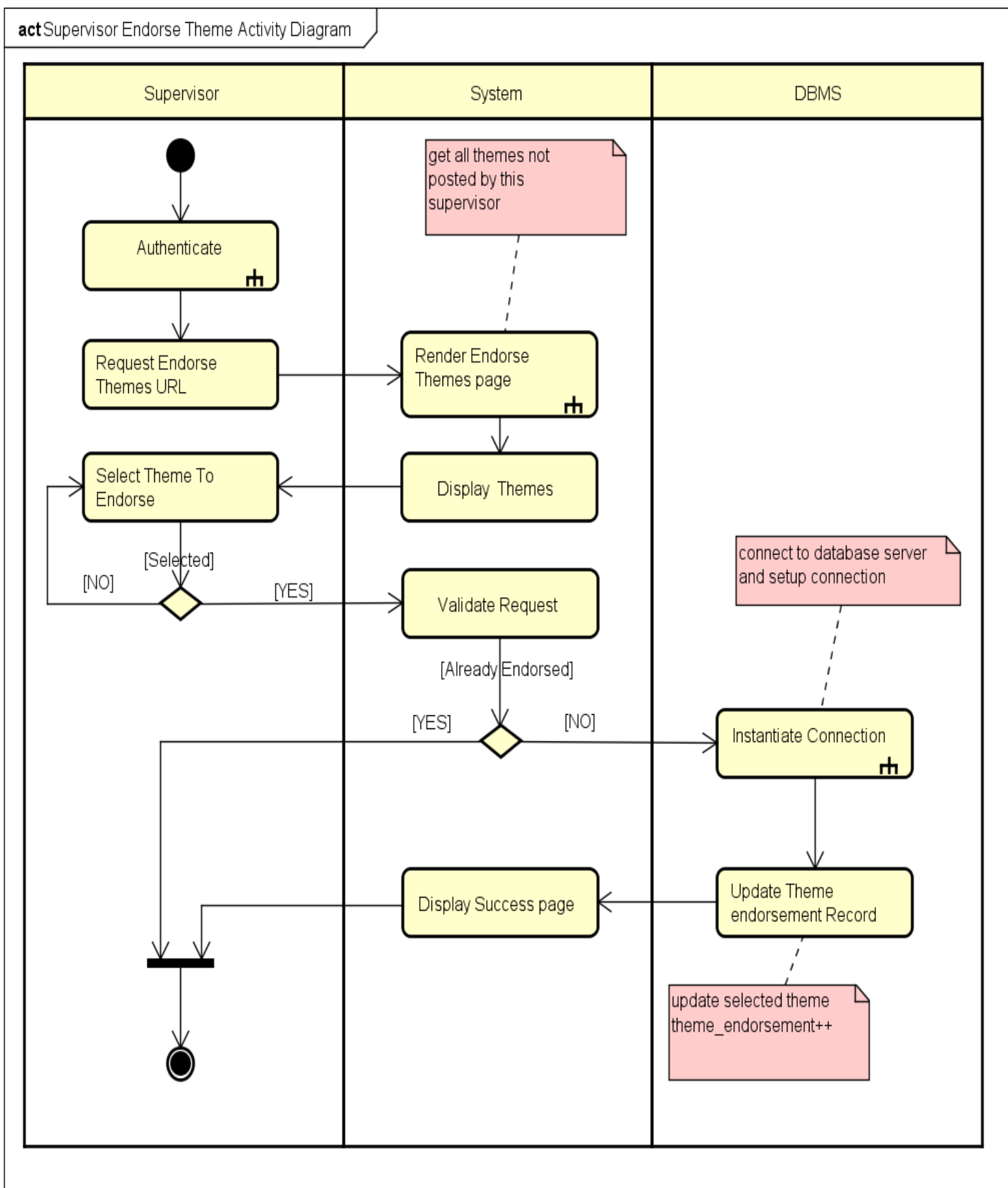


Figure 2.8: Supervisor Endorse Theme Activity Diagram

2.3.5.8 Reviewer Validate Theme Activity Diagram

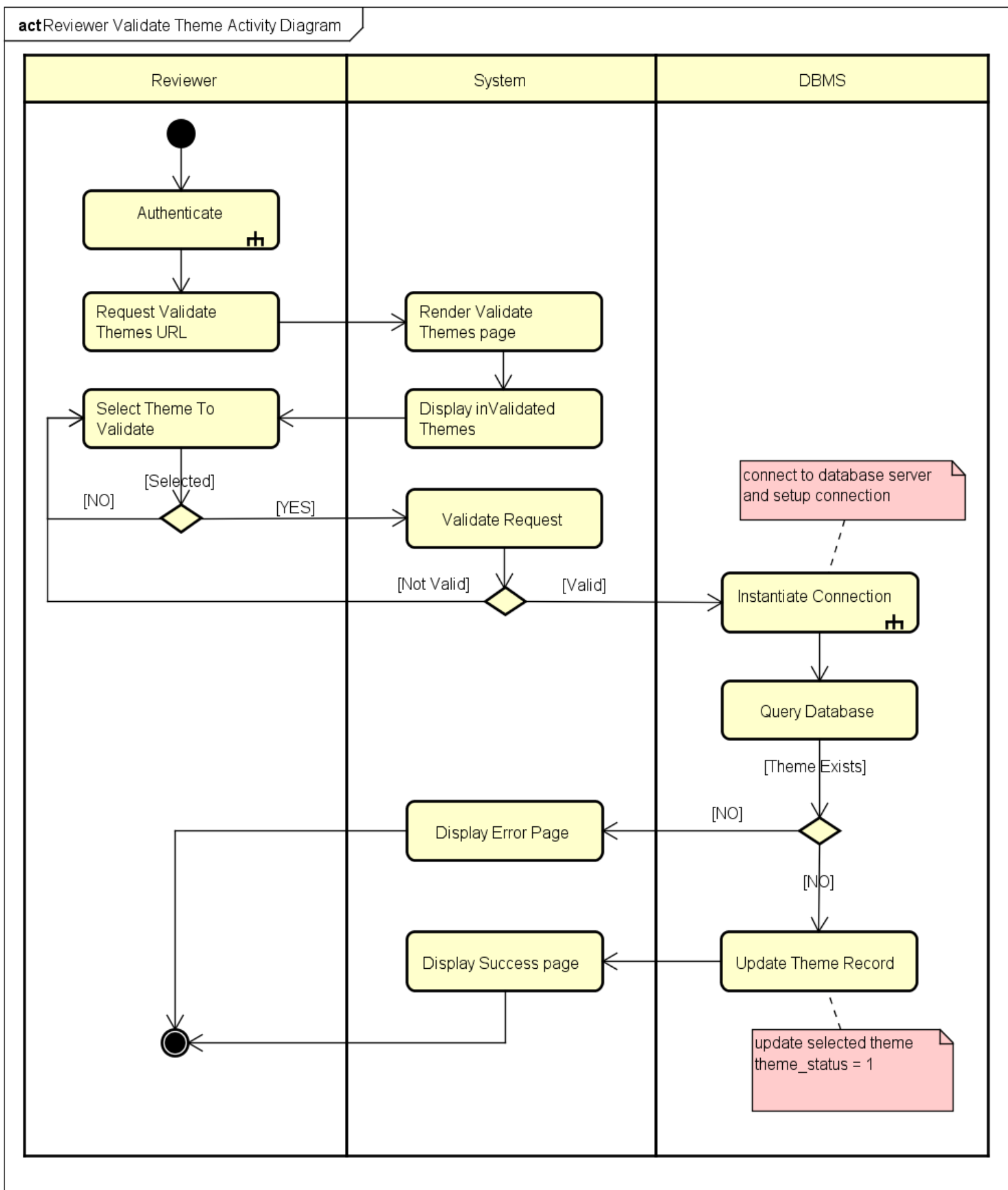


Figure 2.9: Reviewer Validate Theme Activity Diagram

2.3.6 Sequence Diagram

2.3.6.1 Definition

A sequence diagram is an interaction diagram that details how operations are carried out: what messages are sent and when. Sequence diagrams are organized according to time. The time progresses as you go down the page. The objects involved in the operation are listed from left to right according to when they take part in the message sequence. Purpose Of Sequence Diagrams The purposes of sequence diagrams are to visualize the interactive behaviour of the system. Now visualizing interaction is a difficult task. So the solution is to use different types of models to capture the different aspects of the interaction. So the purposes of interaction diagram can be describes as:

- To capture dynamic behaviour of a system.
- To describe the message flow in the system.
- To describe structural organization of the objects.
- To describe interaction among objects.

2.3.6.2 Authentication Sequence Diagram

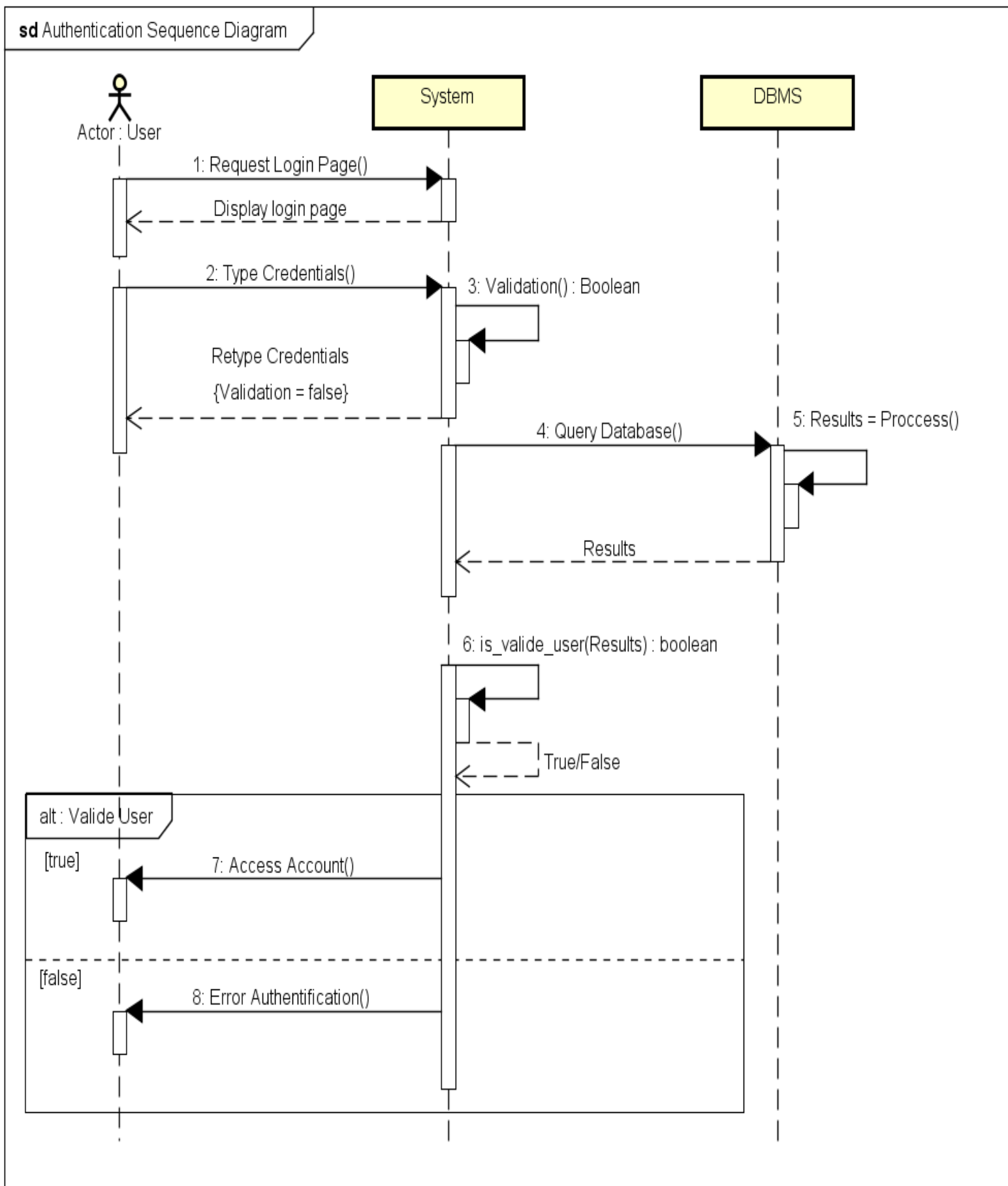


Figure 2.10: Authentication Sequence Diagram

2.3.6.3 Student Register Sequence Diagram

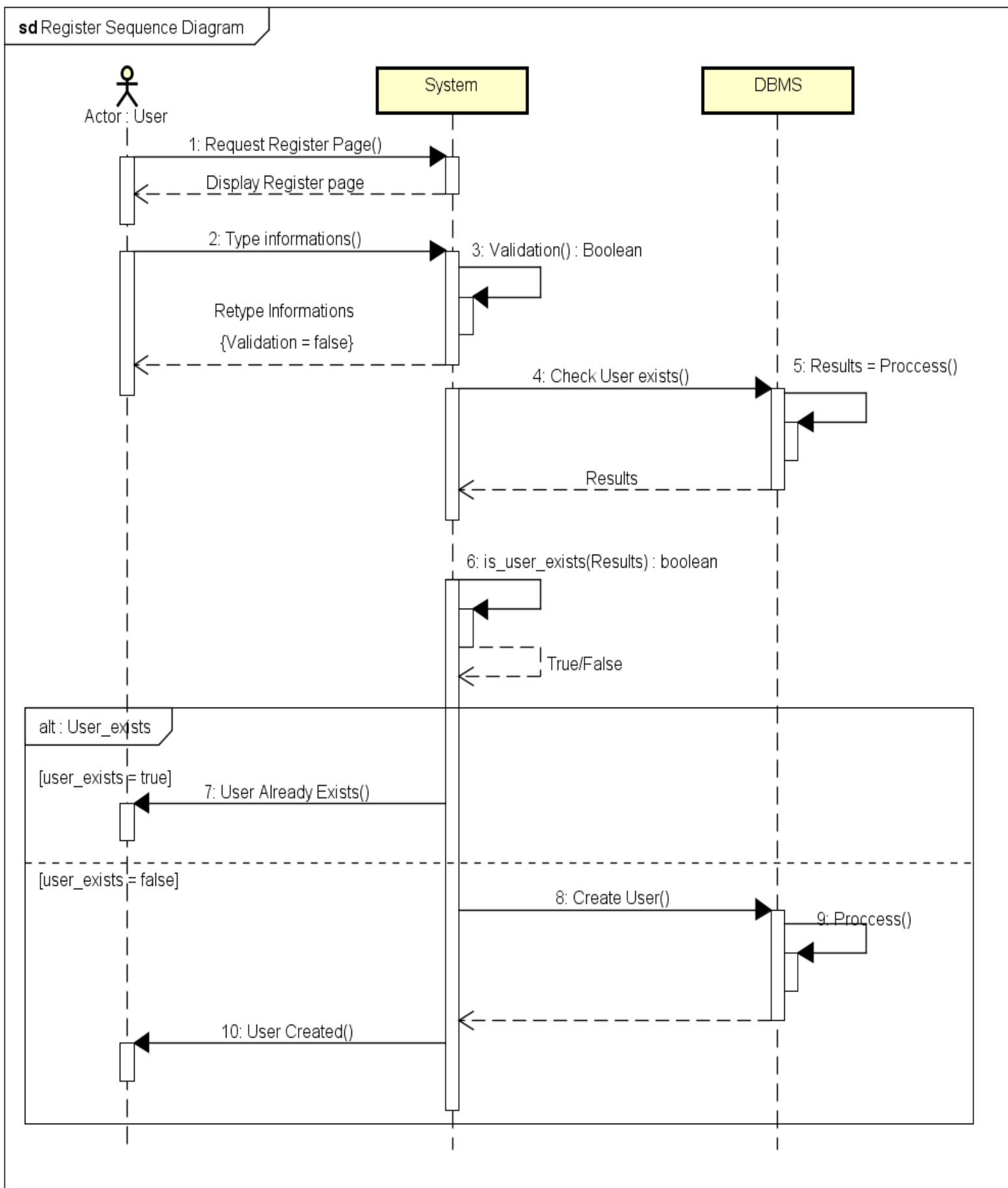


Figure 2.11: Student Register Sequence Diagram

2.3.6.4 Supervisor Post Theme Sequence Diagram

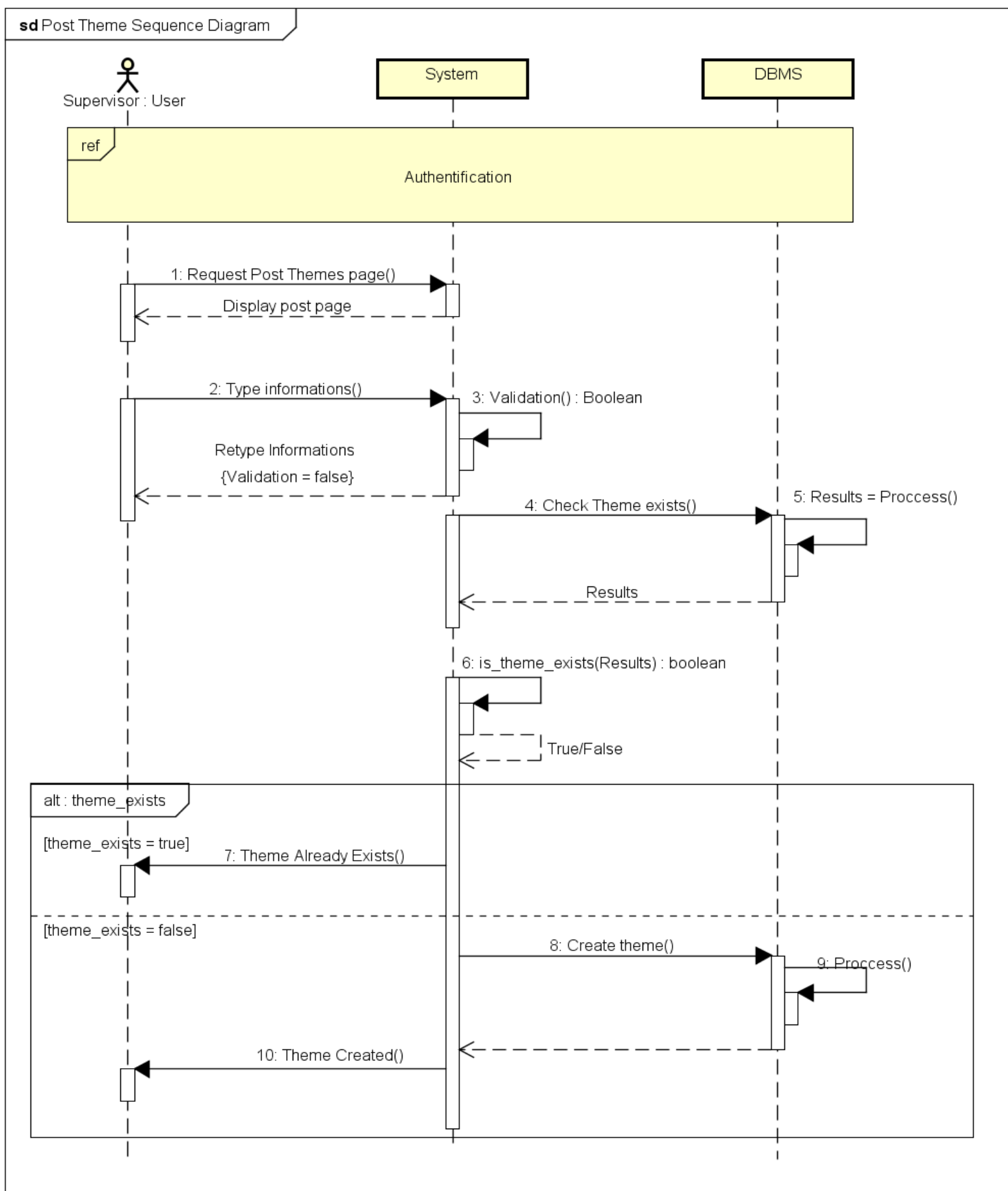


Figure 2.12: Supervisor Post theme Sequence Diagram

2.3.6.5 Supervisor Endorse Theme Sequence Diagram

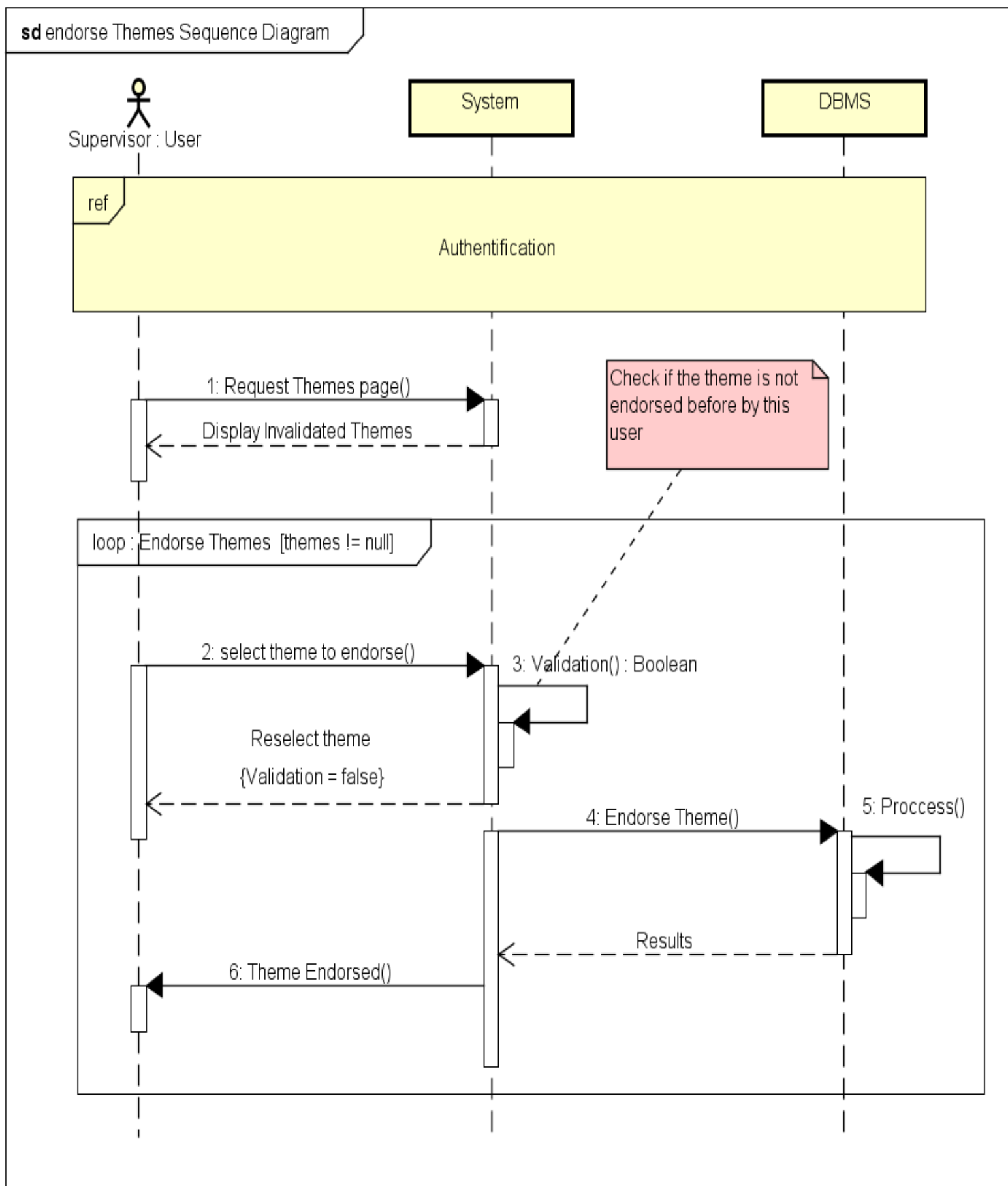


Figure 2.13: Supervisor Endorse Theme Sequence Diagram

2.3.6.6 Reviewer Validate Theme Sequence Diagram

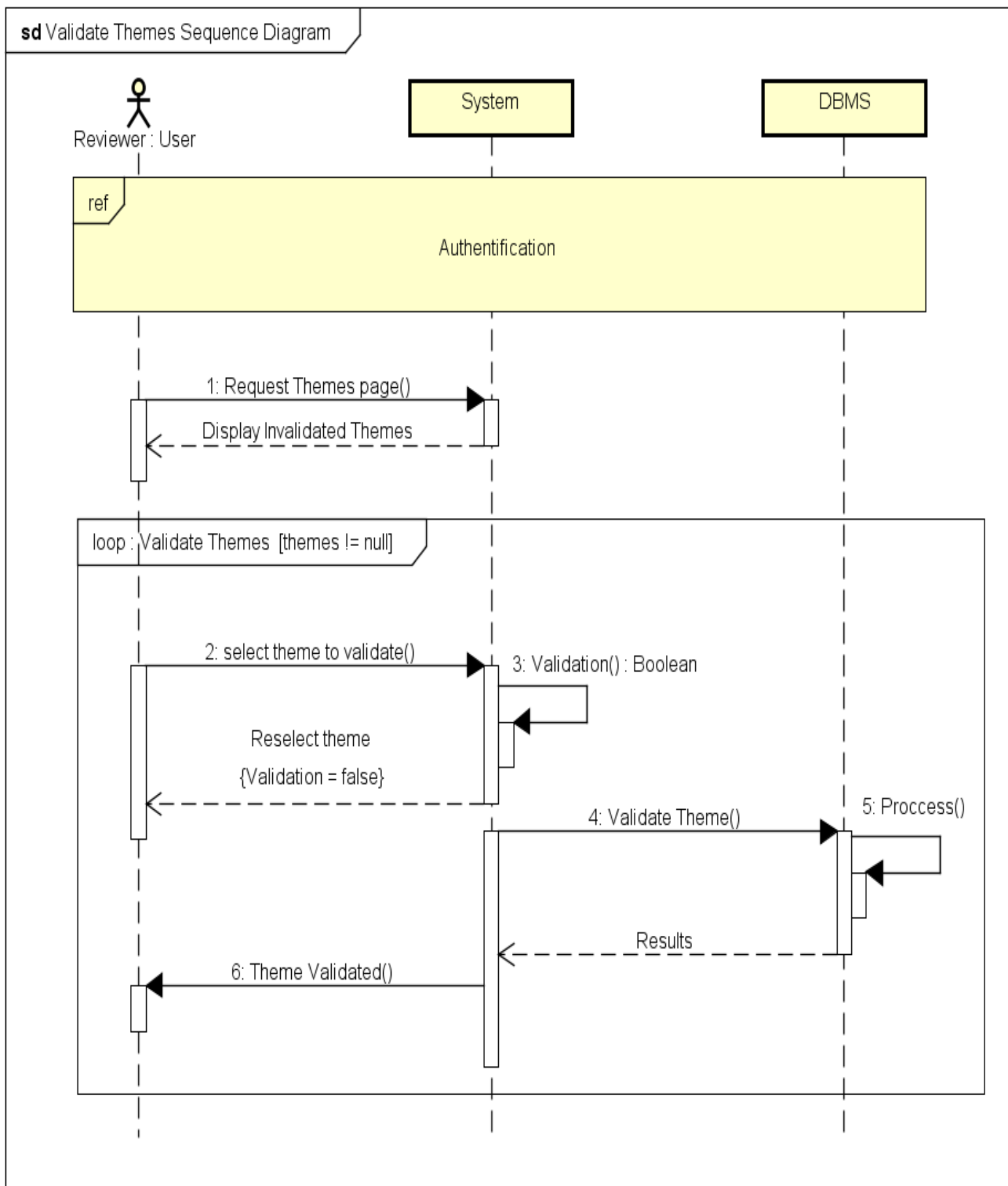


Figure 2.14: Reviewer Validate Theme Sequence Diagram

2.4 Data Modelling

2.4.1 Entity Relationship Diagram (ERD)

The entity-relationship (ER) paradigm [Chen, 1976] supports conceptual modelling at a higher level of abstraction and, thus, more efficiently than the relational paradigm. A model developed using the ER paradigm may be automatically and easily mapped to the corresponding relational model because these two paradigms are conceptually close. This section briefly describes the ER paradigm.

2.4.1.1 An Entity

An entity represents a discrete object or a thing that can be distinctly identified in the problem domain.

2.4.1.2 Relationships

A relationship is a conceptual association between entities. For example, John Smith and Computers for Everybody are in a student-attends-course relationship.

2.4.1.3 Attributes

Attributes are named properties of entity or relationship sets. An attribute maps each entity or relationship in a set to a value.

2.4.2 The Cardinality

The cardinality of each side (role) of a relationship set denotes how many entities of the entity set at one side may be related to an entity of the entity set on the other side through that relationship.

2.4.3 IMS ER Diagram

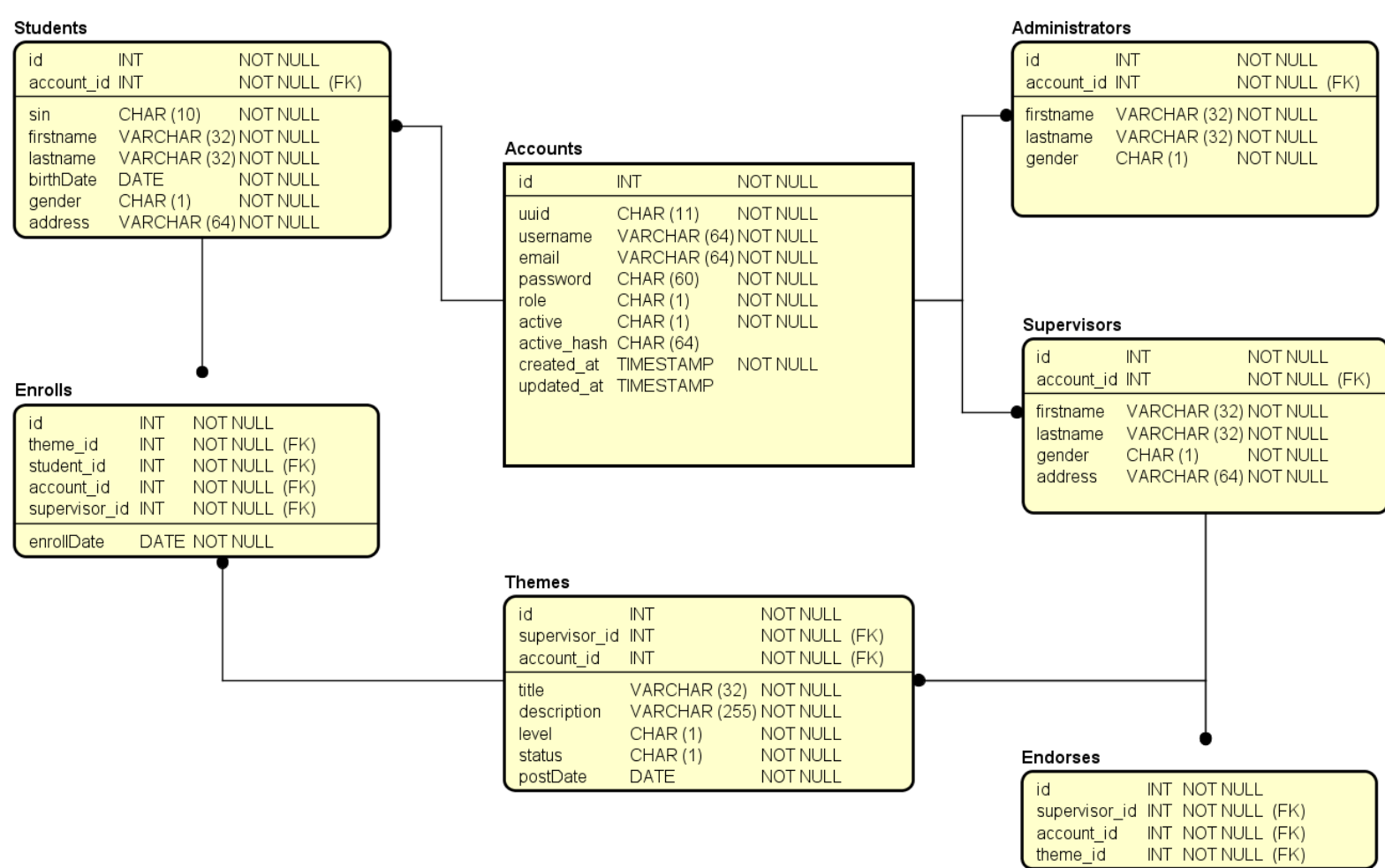


Figure 2.15: IMS ER Diagram

2.5 Architecture Of Implementation

2.5.1 Model, View And Controller

MVC (Model-View-Controller) is a software design pattern built around the interconnection of three main component types, in a programming language such as PHP, often with a strong focus on object-oriented programming (OOP) software paradigms. The three component types are loosely termed models, views, and controllers. Lets talk about them individually and then see how they fit together.

2.5.1.1 Model

The model is where all the business logic of an application is kept. Business logic can be anything specific to how an application stores data, or uses third-party services, in order to fulfil its business requirements. If the application should access information in a database, the code to do that would be kept in the model.

2.5.1.2 View

The view is where all of the user interface elements of our application are kept. This can include our HTML markup, CSS style sheets, and JavaScript files. Anything a user sees or interacts with can be kept in a view, and sometimes what the user sees is actually a combination of many different views in the same request.

2.5.1.3 Controller

The controller is the component that connects models and views together. Controllers isolate the business logic of a model from the user interface elements of a view, and handle how the application will respond to user interaction in the view. Controllers are the first point of entry into this trio of components, because the request is first passed to a controller, which will then instantiate the models and views required to fulfil a request to the application. Shown below is a diagram of the MVC architectural pattern. It shows a typical collaboration of the MVC components:

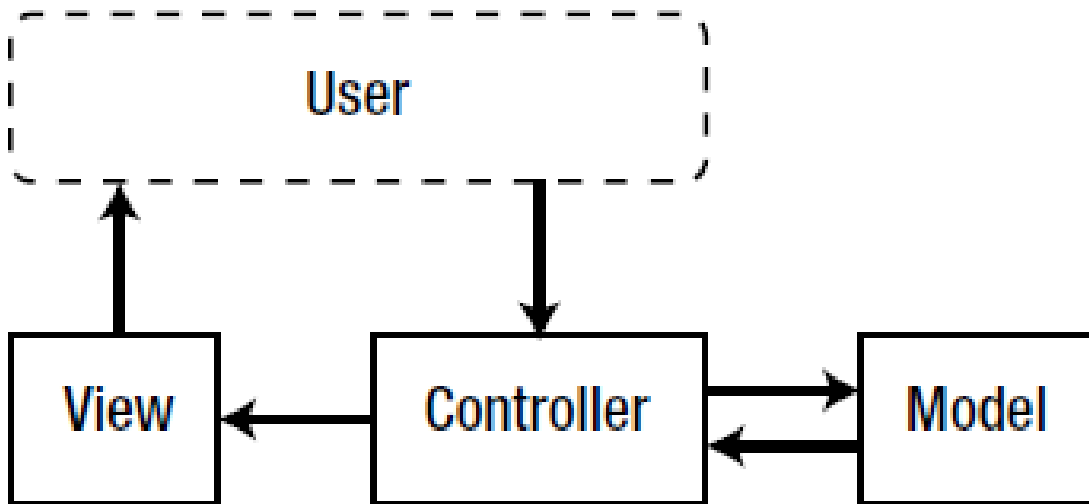


Figure 2.16: MVC Structure Diagram

And the table below describes the advantages and disadvantages of MVC architectural pattern.

| Advantages | Disadvantages |
|---|---|
| It allows for code reusability. | Complex to implement and not suitable for smaller applications because it effects performance and design (Sagar, 2009). |
| There is separation of concerns. | |
| enforces loose coupling between the layers. | |

Table 2.6: Advantages And Disadvantages Of MVC Architectural Pattern

2.5.2 Rational For Choosing MVC

The architecture of implementation adopted was the MVC architectural pattern. This pattern was chosen because the application was to be developed using PHP and most PHP frameworks are MVC based which proves the power of MVC and its flexibility and robustness. And the list below shows popular MVC frameworks:

- CodeIgniter

- Zend Framework
- CakePHP
- Laravel

2.5.3 Conclusion

The design phase is a fundamental step for the realization of any project of any scale. and it helps to achieve the implementation of the system. the next step, is to find the ways and tools possible to develop the system , which we will introduce in the next chapter.

Chapter 3

Implementation

Implementation in this context refers to programming. This is the phase that gave birth to the system. The outputs from the analysis and design phase are used as input in this phase. The main activity in this phase is programming.

3.1 Programming Languages and Tools Used

3.1.1 Programming Languages

3.1.1.1 HTML

HTML (Hypertext Markup Language) is the set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page. The markup tells the Web browser how to display a Web page's words and images for the user. The idea behind Hypertext Markup Language (HTML) was born at the European Laboratory for High- Energy Physics (CERN) in Geneva, Switzerland, as early as 1989. One year later, the World Wide Web project was also started there.

3.1.1.2 CSS

Cascading Style Sheets (CSS) is a style sheets language (style language) introduced by W3C. Cascading refers to the process of determining the priority of styling rules. CSS is used to define the presentational semantics of structured documents. It provides control over visual characteristics of

HTML and XHTML documents and their elements. Some typical features are, for example, fonts, colors, backgrounds, margins, borders, and layers.

3.1.1.3 SASS

Sass (Syntactically Awesome StyleSheets) is an extension of CSS that adds power and elegance to the basic language. It allows you to use variables, nested rules, mixins, inline imports, and more, all with a fully CSS compatible syntax. Sass helps keep large stylesheets well organized, and get small stylesheets up and running quickly.

3.1.1.4 Javascript

JavaScript is the programming language of the Web. The overwhelming majority of modern websites use JavaScript, and all modern web browsers on desktops, game consoles, tablets, and smart phones include JavaScript interpreters, making JavaScript the most ubiquitous programming language in history

3.1.1.5 Ajax

Ajax is an acronym for Asynchronous JavaScript and XML. It is not a programming language but a group of web development technologies related to each other, such as HTML, CSS, DOM, JavaScript, XML, and XSLT. Ajax can be used on the client side to create interactive web applications. Web site applications empowered with Ajax can send data to and retrieve data from servers asynchronously . Ajax is suitable for avoiding full-page reloads when exchanging data asynchronously.

3.1.1.6 PHP

PHP stands for Hypertext Preprocessor. It is a server-side scripting language that powers some of the most popular websites in the world, including WordPress and Facebook. It is open source, relatively easy to learn, and works perfectly with MySQL, making it a popular choice for web developers.

3.1.1.7 MYSQL

Every web application, howsoever simple or complicated, requires a database for storing collected data. MySQL, which is an open source relational database management system. Information in a MySQL database is stored in the form of related tables. MySQL databases are typically used for web application development (often accessed using PHP). A MySQL database can be accessed (queried) directly using; C, C++, Eiffel, Java, Perl, PHP and Python computer languages. There is also third party support for connection in a Windows environment. MySQL databases are queried using a subset of the standard Structured Query Language (SQL) commands.

3.1.2 Tools

3.1.2.1 XAMPP

XAMPP stands for Cross-Platform (X), Apache (A), MySQL (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing purposes. Everything you need to set up a web server application (Apache), database (MySQL), and scripting language (PHP) is included in a simple extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server is extremely easy as well.

3.1.2.2 Git

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows. available for free at(<https://git-scm.com/>)

3.1.2.3 Astah Professional

Astah Professional is a system design tool that supports UML (Unified Modeling Language)

3.1.2.4 Sublime Text 3

Sublime Text is a cross-platform source code editor with a Python application programming interface (API). It natively supports many programming languages and markup languages, and its functionality can be extended by users with plugins, typically community-built and maintained under free-software licenses.

3.1.2.5 Google Chrome browser

Google Chrome browser is an open source program for accessing the World Wide Web and running Web-based applications. The Google Chrome Web browser is based on the open source Chromium project. Google released Chrome in 2008 and issues several updates a year. It is available for Windows, Mac OS X, Linux, Android and iOS operating systems.