

KNOCKING-NEIGHBORS-DOOR



DESIGN DOCUMENT

GROUP 2:

- ***Aghiles AZZOU,***
- ***Zineb EL-AMRANI,***
- ***Duramana KALUMVUATI,***
- ***Ismail LARSANOV,é***
- ***NGUEMO DOUANYA,***
- ***Mickael PIRES,***
- ***Mawuse ALOGNIKOU.***

Summary

INTRODUCTION AND CONTEXT OF THE PROJECT	5
CHALLENGES AND OBJECTIVES OF THE PROJECT	5
I- Introduction to the Unified Modeling Language (UML)	5
II- Presentation of the modeling tool.....	6
III- Detailed study of the solution.....	7
CONCLUSION	22

Table of figure

Figure 1: The logo of UML.....	5
Figure 2: Draw.io logo.....	6
Figure 3: Context-diagram-KNOD.....	9
Figure 4: Visitor use case - KNOD.....	10
Figure 5: Users use case – KNOD	11
Figure 6: Webmasters use case - KNOD	12
Figure 7: Administrator use case – KNOD	13
Figure 8: Account creation activity diagram.....	14
Figure 9: CRUD content activity diagram.....	15
Figure 10: Sign-in activity diagram.....	16
Figure 11: Content consulting activity diagram.....	17
Figure 12: Class diagram - KNOD	19
Figure 13: MCD diagram – KNOD	20
Figure 14: MLD diagram - KNOD.....	21

Table of table

Table 1: Actors of our system.....	7
------------------------------------	---

INTRODUCTION AND CONTEXT OF THE PROJECT

KNOD which stands for **Knocking Neighbors Door** is a project of building an event application that proposes socio-cultural activities to face some problems that destroyed the life of nowadays generation: isolation, the difficulty of integration, social distancing, addiction to social networks. The said social-cultural activities have the form of mini-challenges such as sharing your museum, your restaurant, your jogging, your ice cream, your cinema, etc... and other several activities.

CHALLENGES AND OBJECTIVES OF THE PROJECT

The technological advances of our days provide a certain comfort, especially through social networks where young people prefer to express themselves and neglect the physical relationships that are the most vital. It is with this in mind that KNOD has set itself the challenge of reducing to a relatively high percentage the amount of time spent on social networks and encouraging young people to privilege physical encounters.

Moreover, the challenge for society is mainly social-cultural. KNOD will allow young people to communicate more physically and, therefore, more vital, which will remove the fear of the unknown that is harmful to the development of human relationships.

I- Introduction to the Unified Modeling Language (UML)

The Unified Modeling Language is a graphical modeling language based on pictograms (stylized figurative drawings with a sign function) designed to provide a standardized method for visualizing the design and architecture of a system.



Figure 1: The logo of UML

In addition, UML models the system according to three modes of representation:

- ❖ Structural or static diagrams:
 - The class diagram ;
 - The object diagram ;
 - The component diagram;
 - The deployment diagram;

- The composite structure diagram ;
 - The package diagram ;
 - The profile diagram.
- ❖ Behavioral diagrams:
- The use case diagram;
 - Activity diagram;
 - The state-transition diagram.
- ❖ Interaction or dynamic diagrams:
- Sequence diagrams ;
 - The communication diagram;
 - The global interaction diagram;
 - The timing diagram.

These three modes of representation are necessary and complementary to diagram the way the system is composed and how its components work together.

II- Presentation of the modeling tool

Draw.io is a free and open-source cross-platform graph drawing software developed in HTML5 and JavaScript. Its interface can be used to create diagrams such as flowcharts, wireframes, UML diagrams, organizational charts, and network diagrams. Draw.io is available as an online cross-browser web app, and as an offline desktop application for Linux, macOS, and Windows. Its offline application is built using the Electron framework. The web app does not require online login or registration and can open from and save to the local hard drive. Supported storage and export formats to download include PNG, JPEG, SVG, and PDF. It also integrates with cloud services for storage including Dropbox, OneDrive, Google Drive, GitHub, and GitLab.com. It is also available as a plugin to embed the web app in platforms such as NextCloud, MediaWiki, Notion, Atlassian Confluence, and JIRA. It has been described by tech reviewers such as TechRadar and PCMag as an alternative to Lucidchart, Microsoft Visio, SmartDraw, and Cacoo.



Figure 2: Draw.io logo

III- Detailed study of the solution

In this section, the different UML diagrams will be represented according to the client's needs.

1- Preliminary study

We detail in this sub-section the actors of the future system through the context diagram.

1.1- Context diagram

This diagram is not officially designated as a UML diagram. It is therefore not one of the 13 "official" diagrams, but it is useful for defining the actors, before starting to look at other aspects, such as packages and use cases.

- **What are actors?**

According to << UML2 by Pascal ROQUES Practice >>, an actor represents a role played by an external entity (human user, hardware device, or another system) that interacts directly with the system. The actors of our system are grouped in the following table 1:

Table 1: Actors of our system

ACTORS	TYPES
Visitors	PRINCIPAL
Users (Subscribers of our platform)	
Administrators	
Webmaster	

Therefore here is the context diagram of our system:

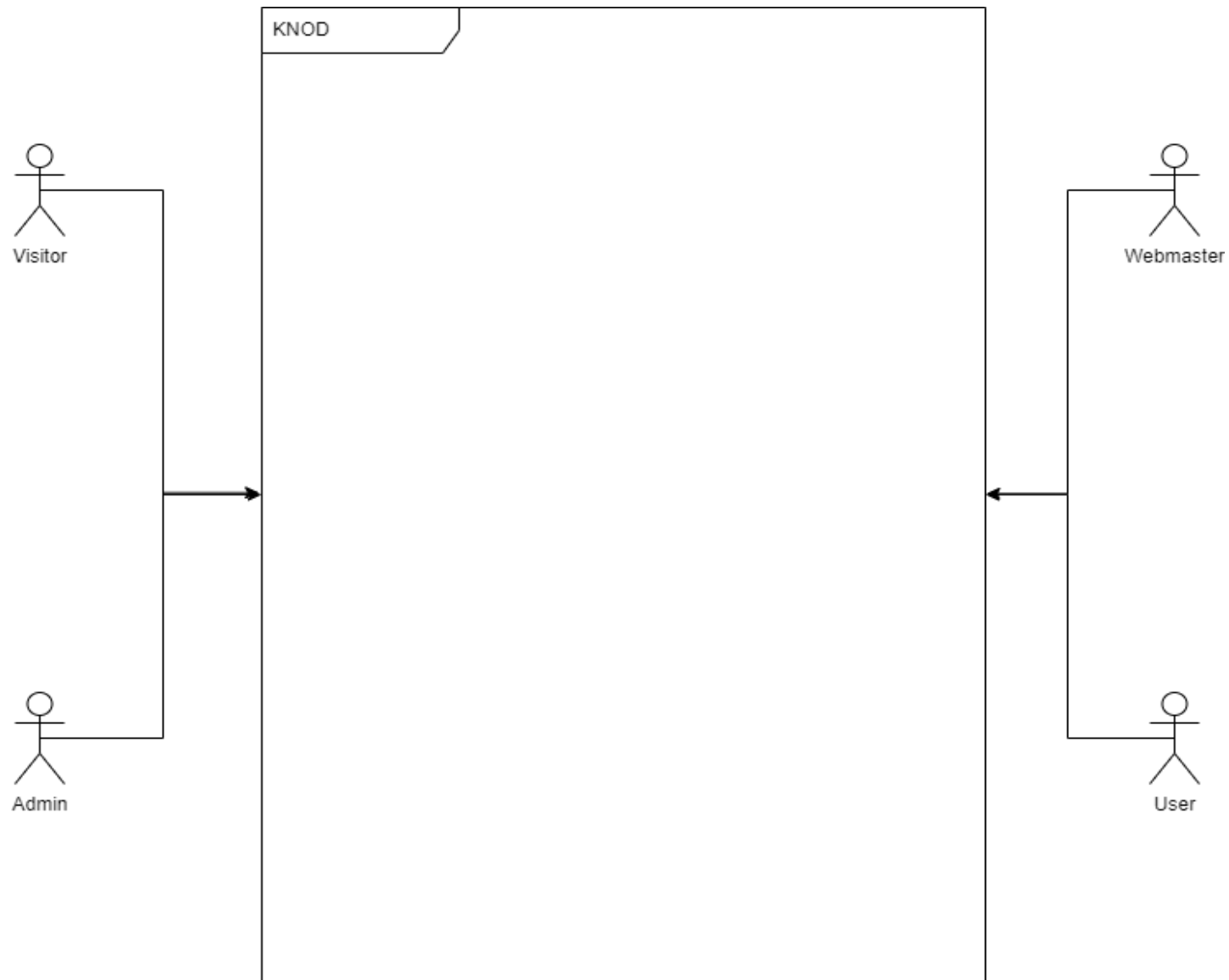


Figure 3: Context-diagram-KNOD

2- Functional requirements

In this sub-section, we detail the functional specifications of the future system through diagrams such as use case, activity, and system sequence diagrams.

2.1- Use case diagram

The use case diagram is a diagram that shows the use cases (ovals) connected by associations (lines) to actors (stick man icon or equivalent graphic representation).

➤ Use cases:

A use case is a description of the interactions that will allow the actor to reach his goal by using the system.

➤ Use cases diagram representation:

For clarity reason, we've decided to group use case per actor.

- **Visitors use case:**

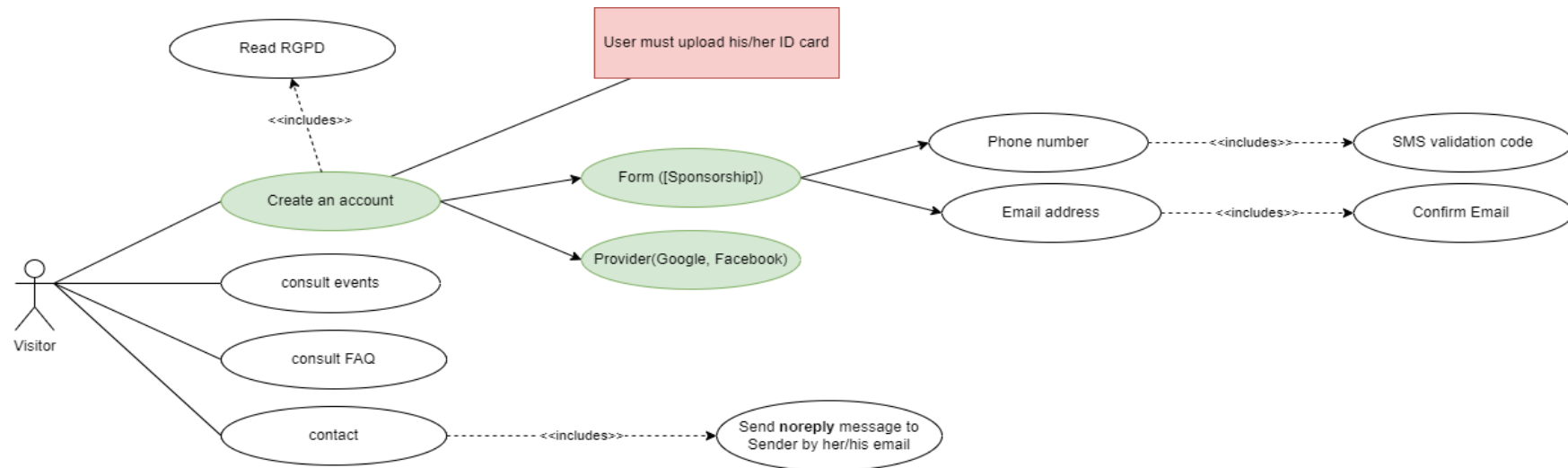


Figure 4: Visitor use case - KNOD

- **Users use case:**

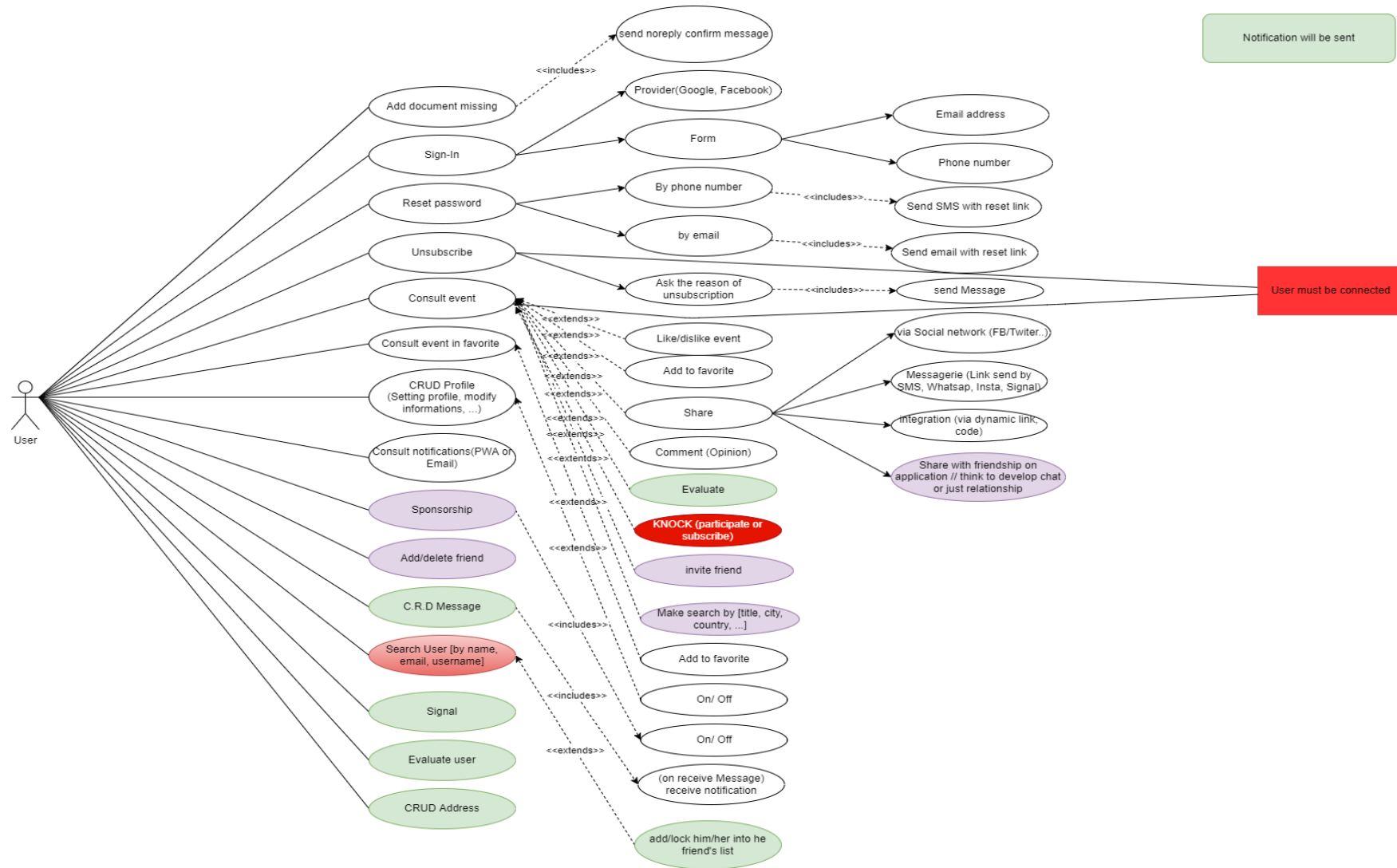


Figure 5: Users use case – KNOD

- **Webmasters use case:**

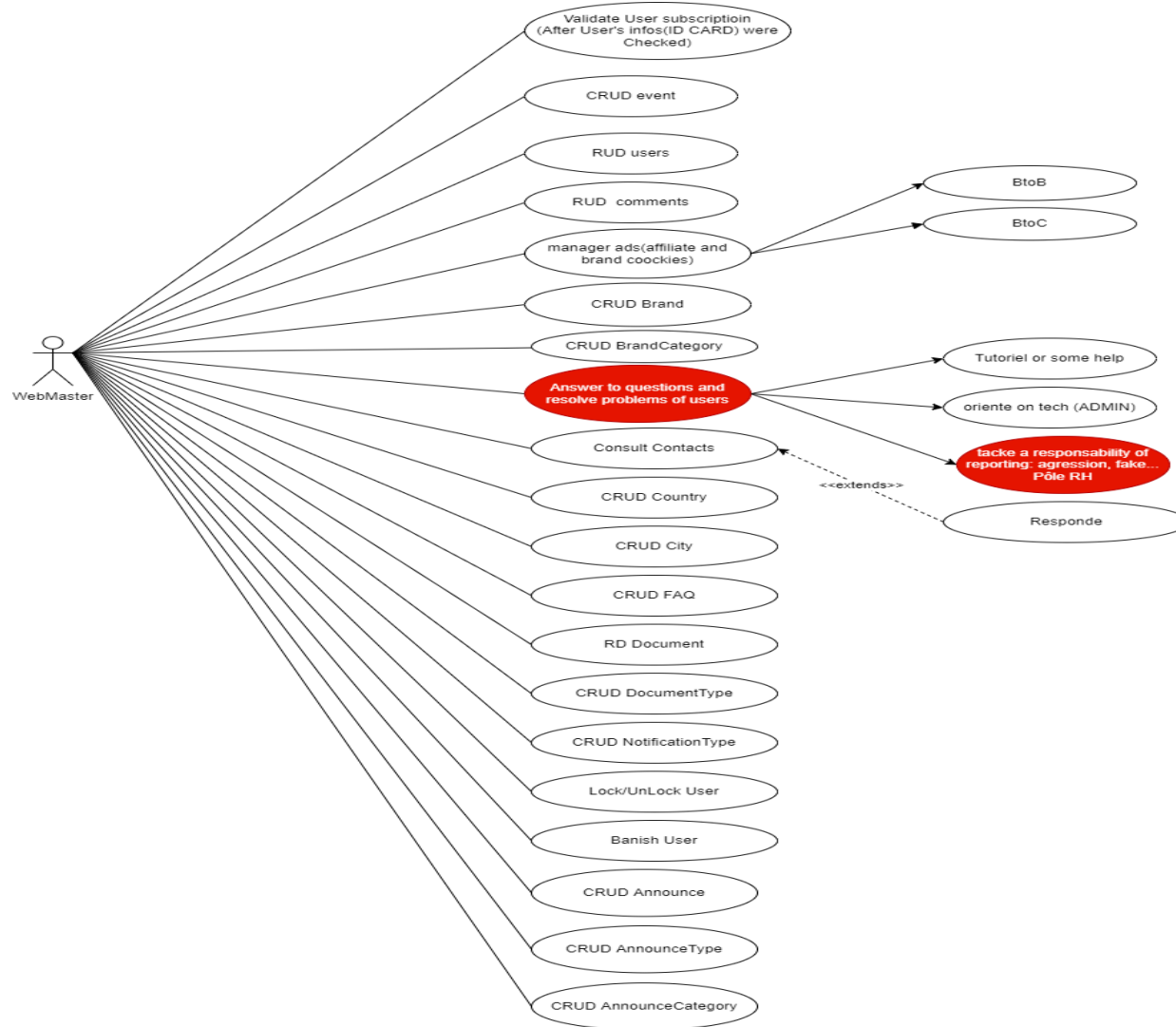


Figure 6: Webmasters use case - KNOD

- **Administrators use case:**

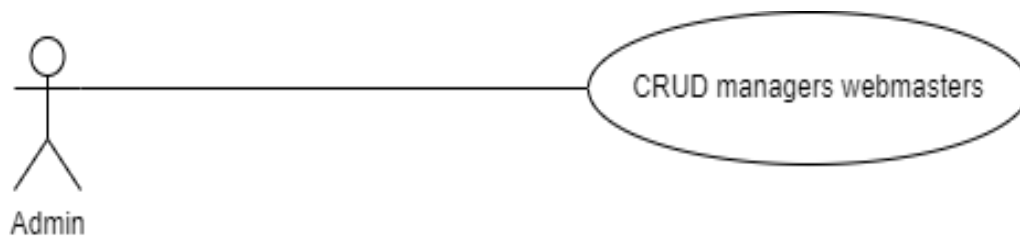


Figure 7: Administrator use case – KNOD

2.1- Activity diagram

In UML, the activity diagram is a diagram that allows to represent the triggering of events according to the system states and to model the behaviors. It is a flowchart that describes the sequences or steps in the development of a use case or a set of use cases. Here are some activity diagrams associated with our project:

• **Account creation activity diagram:**

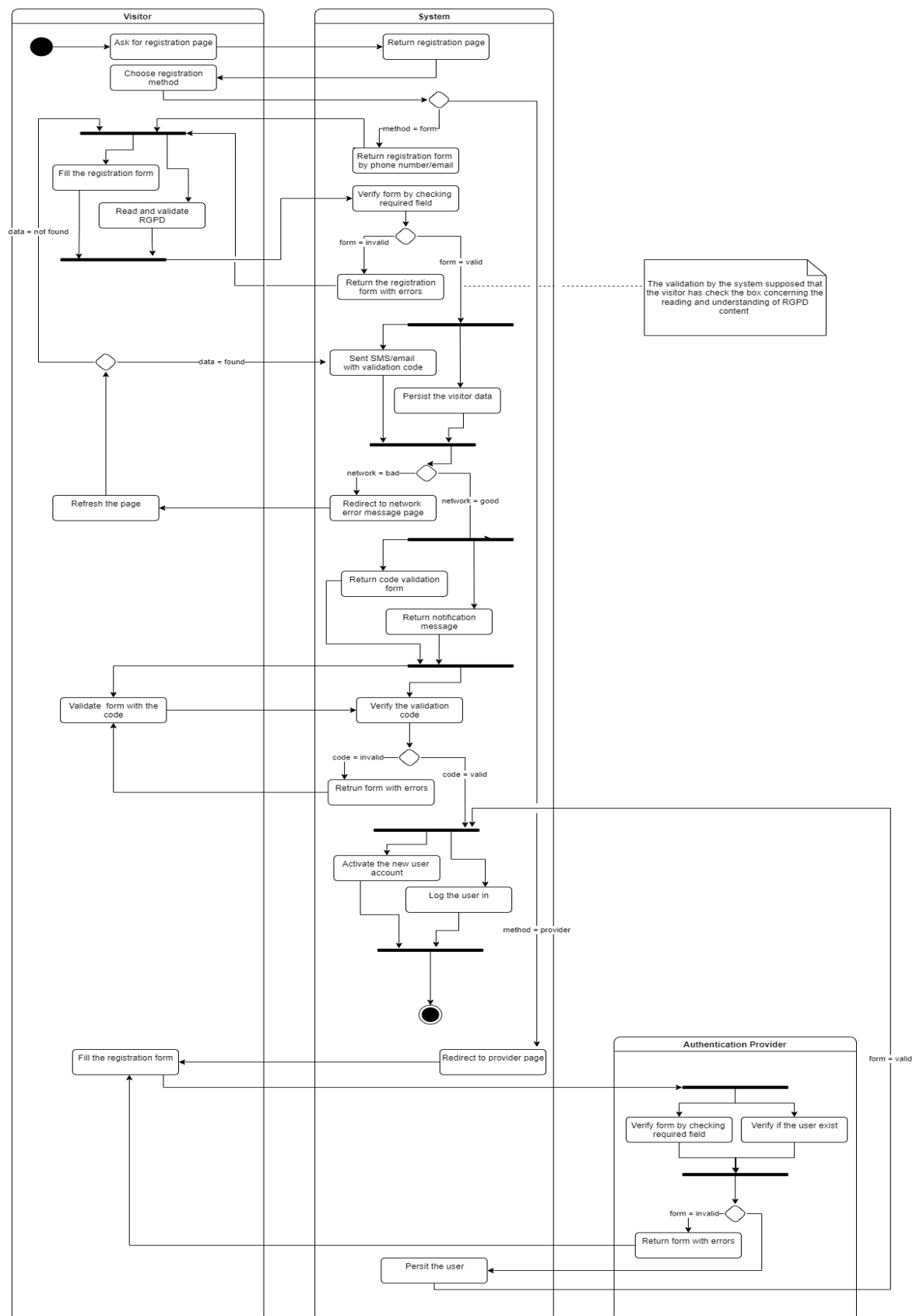


Figure 8: Account creation activity diagram

• **CRUD content activity diagram:**

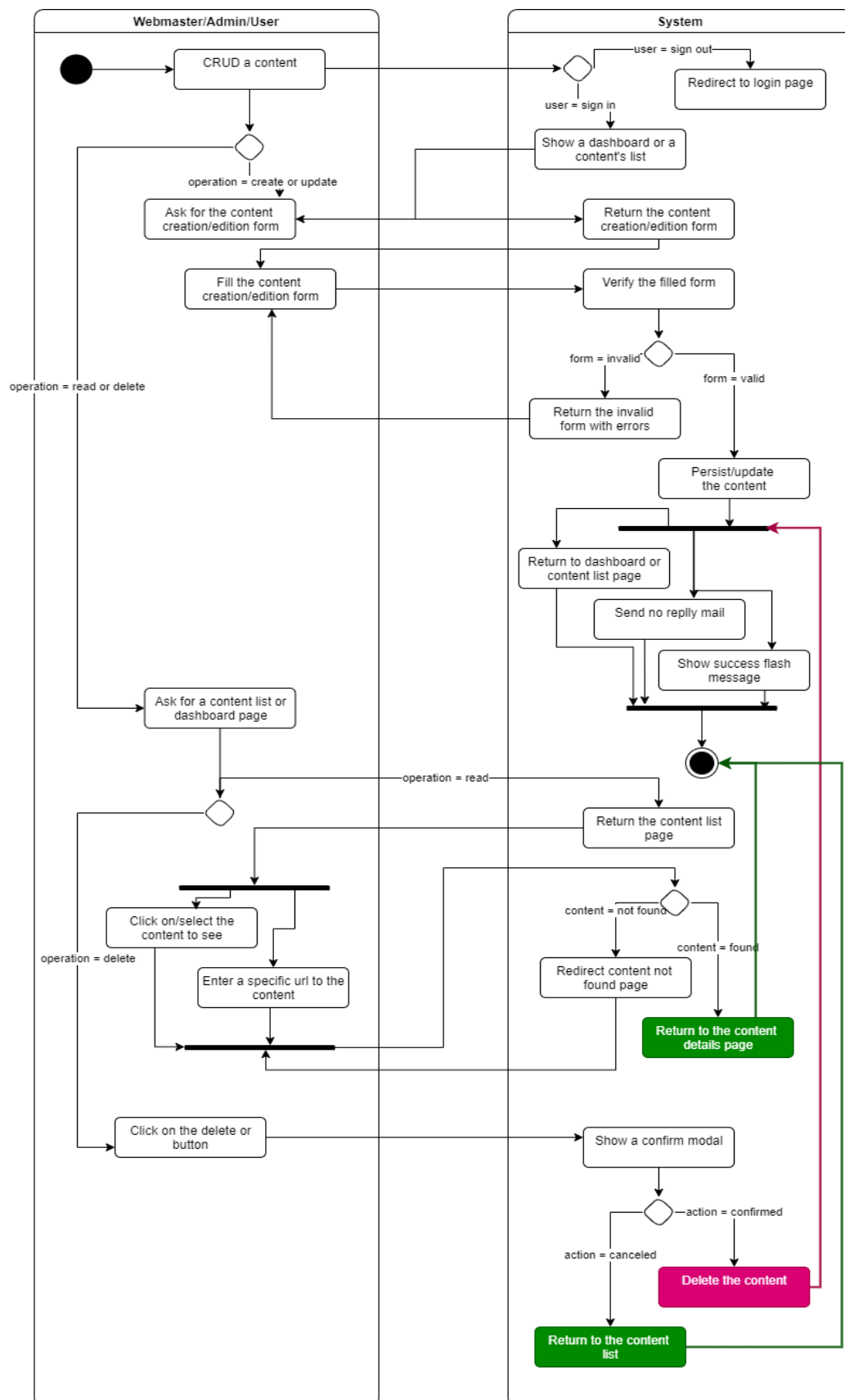


Figure 9: CRUD content activity diagram

- **Sign-in activity diagram:**

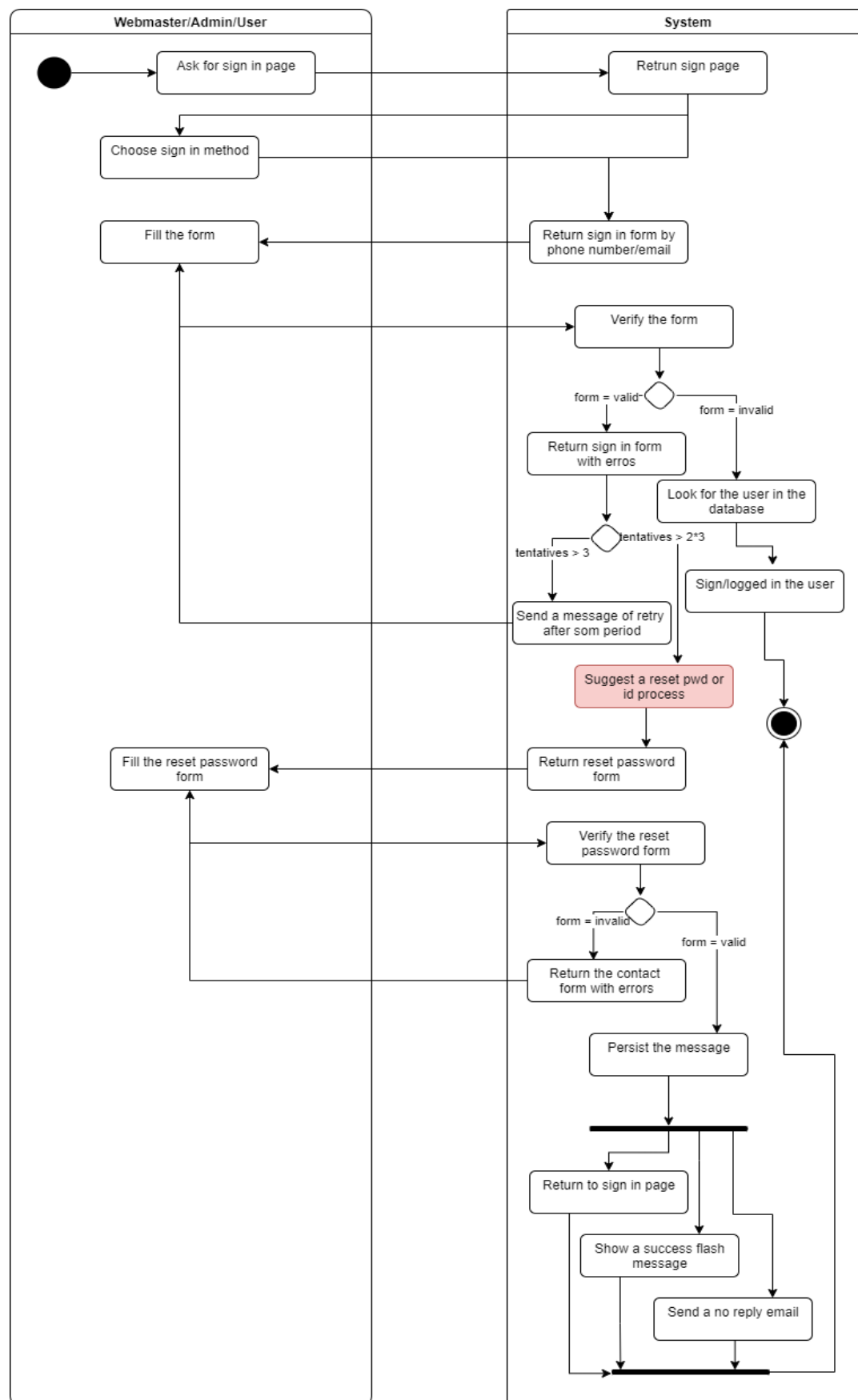


Figure 10: Sign-in activity diagram

- **Content consulting activity diagram:**

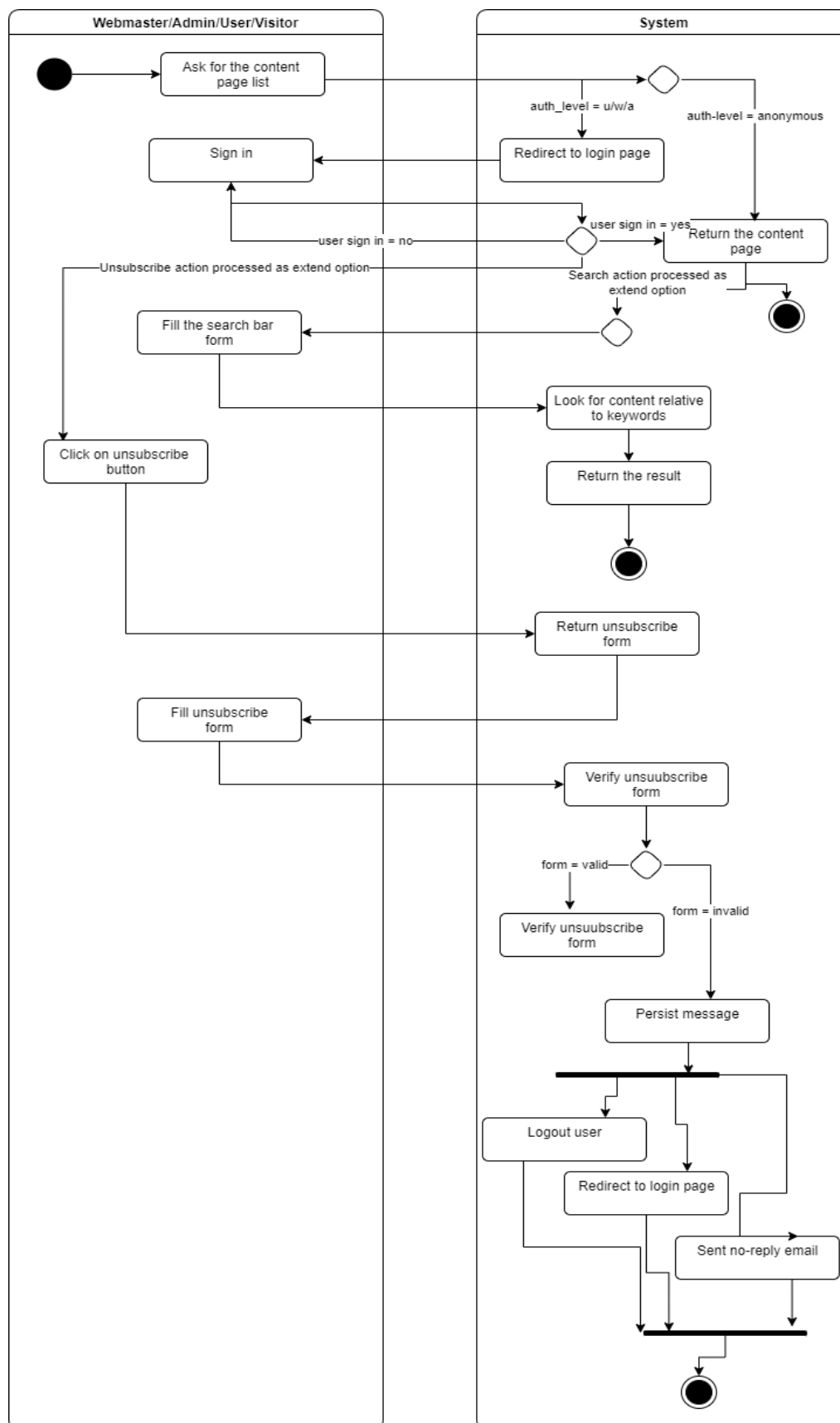


Figure 11: Content consulting activity diagram

3- Analysis and design

After the description of the requirements using the use case diagram and of the processes using the sequence diagram detailed in the activity diagram, we can model the logical structure of the system, i.e. its static aspects. This modeling is largely done in class diagrams, possibly with diagrams, with possibly object diagrams showing the specific configurations of the system under particular conditions. The main content of this section is therefore the presence of the modeling elements of the class and deployment diagram to describe the structure of our application.

3.1- Class diagram

The class diagram is considered to be the most important in object-oriented modeling; it is the only mandatory one in such modeling. It is a diagram presenting the classes and interfaces of the system as well as the different relationships between them. While the use case diagram shows a system from the actors' point of view the class diagram shows the internal structure of the system. It provides an abstract representation of the objects of the system that will interact to achieve the realized use cases. This diagram is part of a static branch of The class diagram is part of the static branch of the UML because it abstracts the temporal and dynamic aspects. A class describes the responsibilities, behavior, and type of a set of objects. The elements of this set are the instances of the class. The class diagram of our application is as follows:

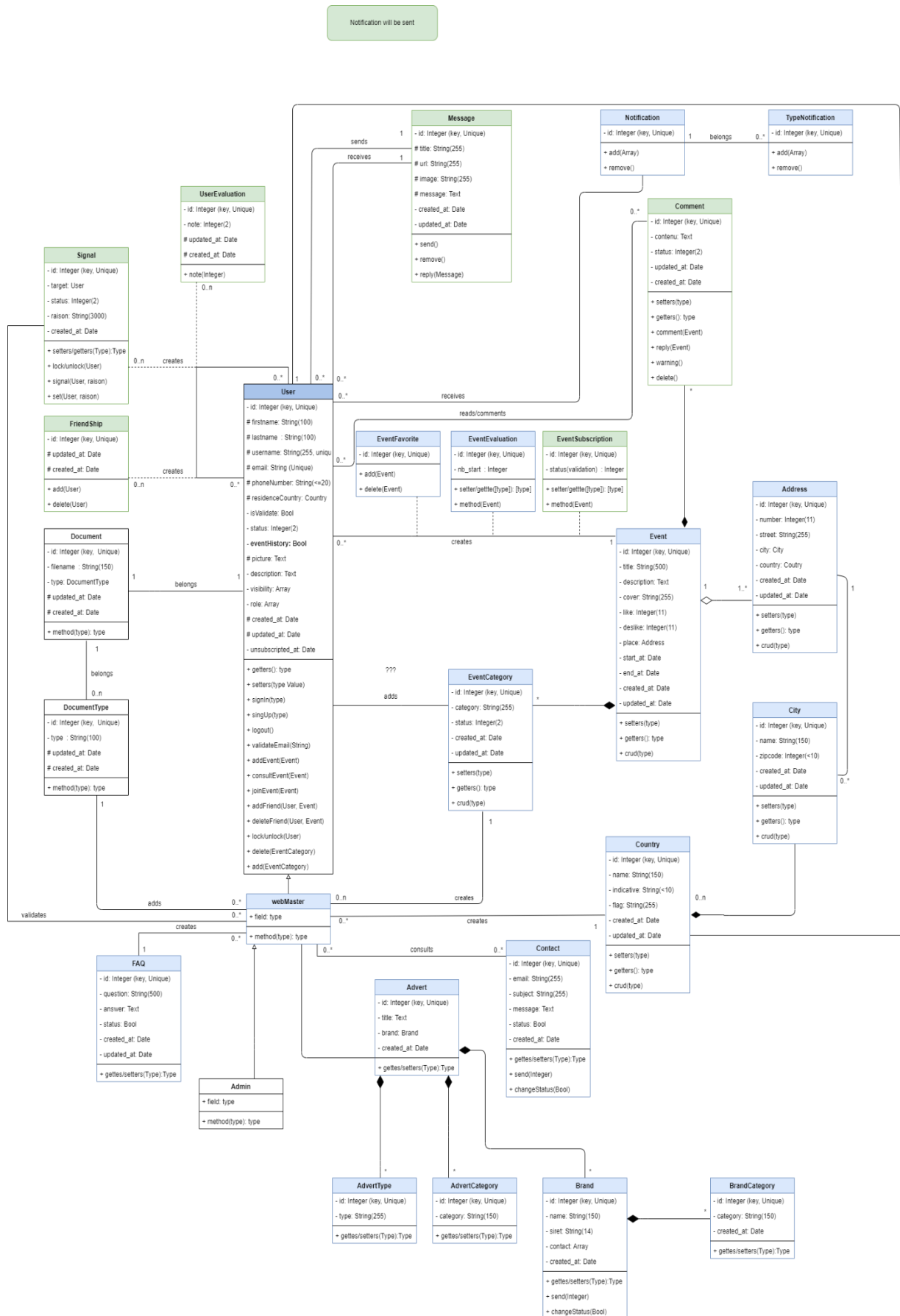


Figure 12: Class diagram - KNOD

Then we deduce the following MLD and MCD diagram:

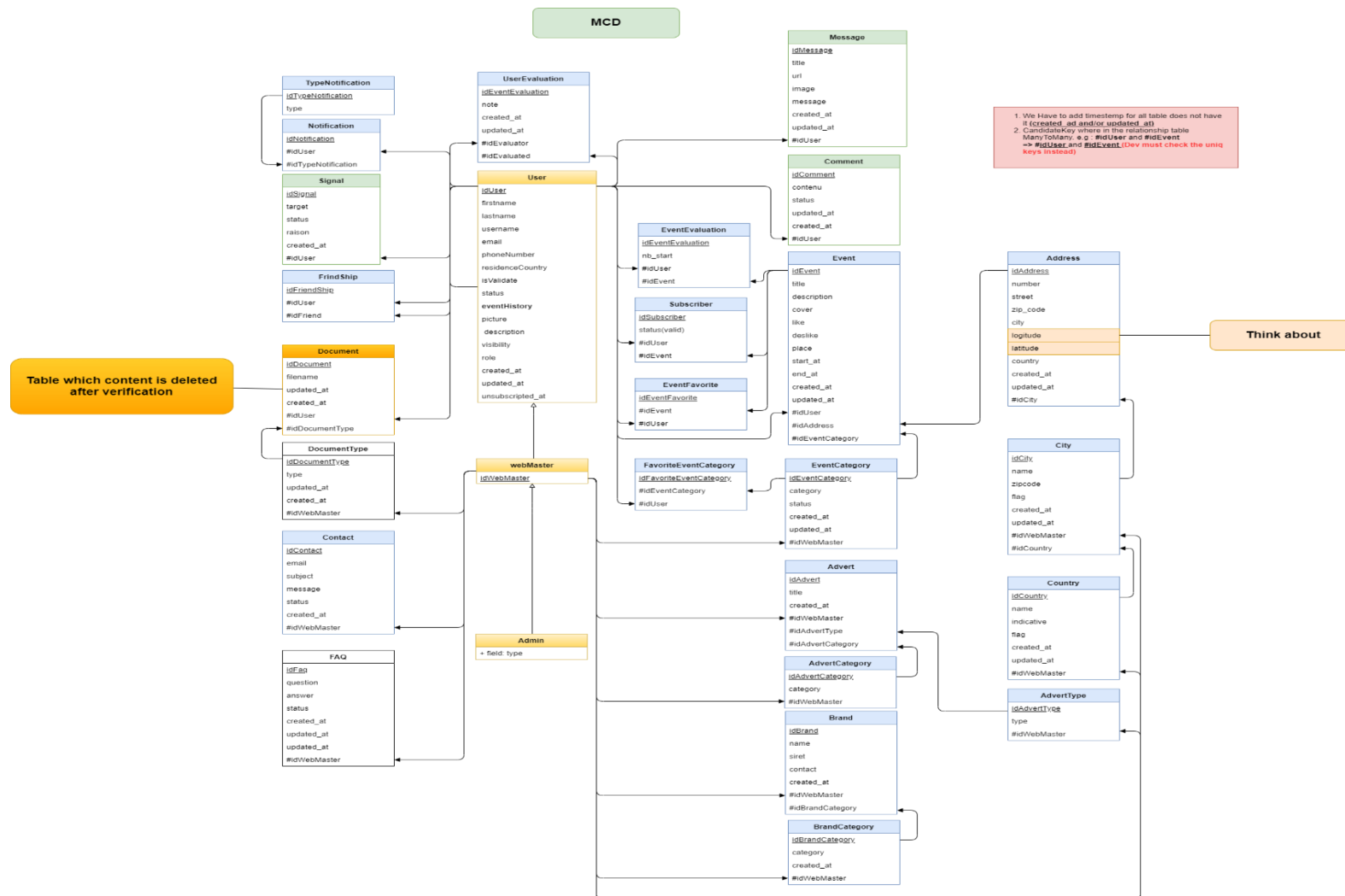


Figure 13: MCD diagram – KNOD



CONCLUSION

Like any type of project, an IT project requires an analysis phase, followed by a design phase. In the analysis phase, we first try to understand and describe precisely the needs of the users or customers. What do they want to do with the software? What features do they want? For what purpose(s)? How should the action work? This is what we call "needs analysis". After validating our understanding of the needs, we imagine the solution. This is the "solution analysis" part. In the design phase, we bring more details to the solution and try to clarify technical aspects such as the installation of the different software parts on the hardware. We will therefore present in this part, the analysis method as well as the modeling tool and a detailed study of the solution.

Table of content

Summary.....	2
Table of figure.....	3
Table of table.....	4
INTRODUCTION AND CONTEXT OF THE PROJECT	5
CHALLENGES AND OBJECTIVES OF THE PROJECT	5
I- Introduction to the Unified Modeling Language (UML)	5
II- Presentation of the modeling tool.....	6
III- Detailed study of the solution.....	7
1- Preliminary study	7
1.1- Context diagram.....	7
2- Functional requirements.....	9
2.1- Use case diagram.....	9
2.1- Activity diagram.....	13
3- Analysis and design.....	18
3.1- Class diagram	18
CONCLUSION	22