Living web application concept, logical functioning and proposed implementation

Fokwa T. Divine, Ngatchu Damen

Department of Computer Engineering, Faculty of Engineering and Technology, University of Buea, P.O box 63, South West Region, Cameroon

Abstract

Over the last decade, there has been an extensive growth in web applications development. Many large scale software systems are now web-based and many software industries are now focused on delivering web-based applications. Base on this increase growth, this paper uses techniques in Unified Modeling language (UML), Rational Unified Process (RUP) and Object Relational Mapping (ORM) to present the concept of a living web application, its logical functioning and a proposed implementation. A living web application is defined as one in which the content is constantly and continually being modified by those who interact with the application. The central piece of the system is in the application's engine. This paper presents this concept using three work flows of RUP, which include: logical modeling, Requirements and, Analysis and Design. This concept is presented in this paper such that it serves as a teaching note for students and to individuals interested in understanding how a combination of these techniques can be applied to present concepts in ICT.

Keywords: Living Web Application; Unified Modeling Language; Rational Unified Process

1. Introduction

Over the last decade, the usage of web-based applications has been increasingly growing. Web applications are becoming more powerful. Examples of such areas where this growth can be clearly seen include the Internet, Database Management Systems, Enterprise Resource Planning Systems etc. Despite this vast growth, little has been presented with respect to a living web application. A living web application is defined as one in which the content is constantly and continually being modified by those who interact with the application. The central piece of the system is in the application's engine.

This paper thus seeks to present this concept, its logical functioning and a proposed implementation with notions drawn from the Unified Modeling Language (UML), Rational Unified Process (RUP) and Object Relational Mapping (ORM), three wide areas in fields such as computer engineering, software engineering, computer science etc.

This paper intends to provide further insights to students and ICT professionals via the usage of a combination technique (UML, RUP and ORM) for presenting the concept of a living web application. The main question that guides this study is the following: What is a living web application and how is it implemented?

This study is presented under the assumption that the reader has prior knowledge of UML and an understanding of the basic diagram types. The results drawn from this paper may enable students and ICT professionals to improve in their analysis of software systems and the exploration of modern techniques to address client's needs.

.

2. Literature Review

Numerous frameworks and languages exist for modeling software applications. The framework and language used in this paper is RUP and UML respectively. This framework and language is used to present the concept of a living web application, its logical functioning and a propose implementation. Object relational mapping is used to show how objects and their relationships are stored in relational databases.

2.1. Rational Unified Process Framework

It is an iterative software development process framework created by the Rational Software Corporation. RUP organizes development of software systems into 4 phases, each consisting of one or more executable iterations of the software at that stage of development (RUP - Rational Unified Process).

The Phases are: Inception, Elaboration, Construction and Transition.

Shown below are the phases and workflows in the rational unified process.

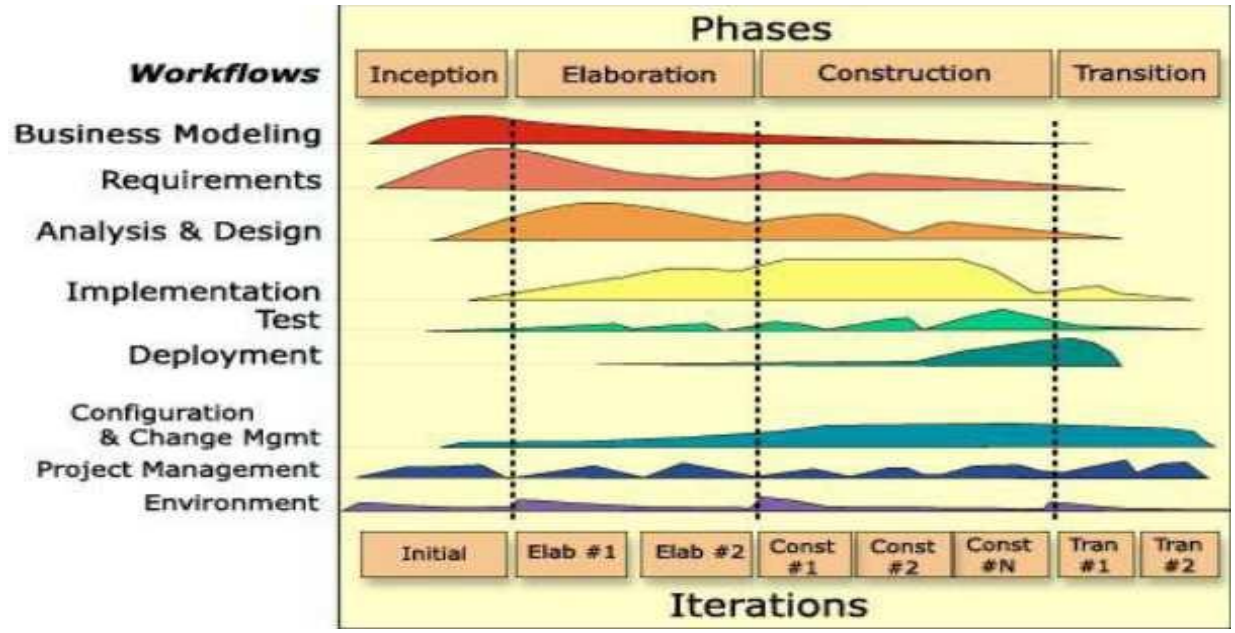

Figure 2.1: RUP Phases and Workflows

2.2. Unified Modeling Language

It is a general purpose modeling language which is designed to provide a standard way to visualize the design of a system (Wikipedia, UML).

UML 2 describes 13 diagrams types listed in table 2.1.

Table 2.1: Official Diagram Types of the UML (Fowler, 2004)

Diagram	**Purpose**
Activity	Procedural and parallel behavior
Class	Class, features, and relationships
Communication	Interaction between objects; emphasis on links
Component	Structure and connections of Components
Composite structure	Runtime decomposition of a class
Deployment	Deployment of artifacts to nodes
Interaction overview	Mix of sequence and activity diagram
Object	Example configurations of instances
Package	Compile-time hierarchic structure
Sequence	Interaction between objects; emphasis on sequence
State machine	How events change an object over time
Timing	Interaction between objects; emphasis on timing
Use Case	How users interact with a system

A subset of these diagrams together with a combination of a subset of RUP workflow is used to present the living web applications concept.

3. Methodology

RUP discerns three workflows where UML is used. This includes:

- Business Modeling
- Requirements
- Analysis and Design

For the purpose of presenting the abovementioned concept, the workflow below will instead be used which deviate slightly from RUP . This include:

- Logical Modeling
- Requirements
- Analysis and Design.

In order to perform logical modeling, UML Conceptual Class diagram, Object diagram and State machine diagram is used. For Requirements, UML Use Case Diagram, Sequence diagram and Activity diagram is used. Lastly, for Analysis and Design, UML Design Class Diagram and Component Diagram is used. Also, a physical data model is provided from a mapping of the design class diagram. All these diagrams together are used to present the abovementioned concept and hence provide an answer to the abovementioned study question.

3.1. Logical Modeling

a. Conceptual Class Diagram

This diagram describes the types of objects in the system and various kinds of static relationships that exist among them. Conceptual class diagram are very useful in exploring the language of a business (Fowler, 2004). That's, it is used to model the information domain. The conceptual class diagram below models the information domain of the living web application.

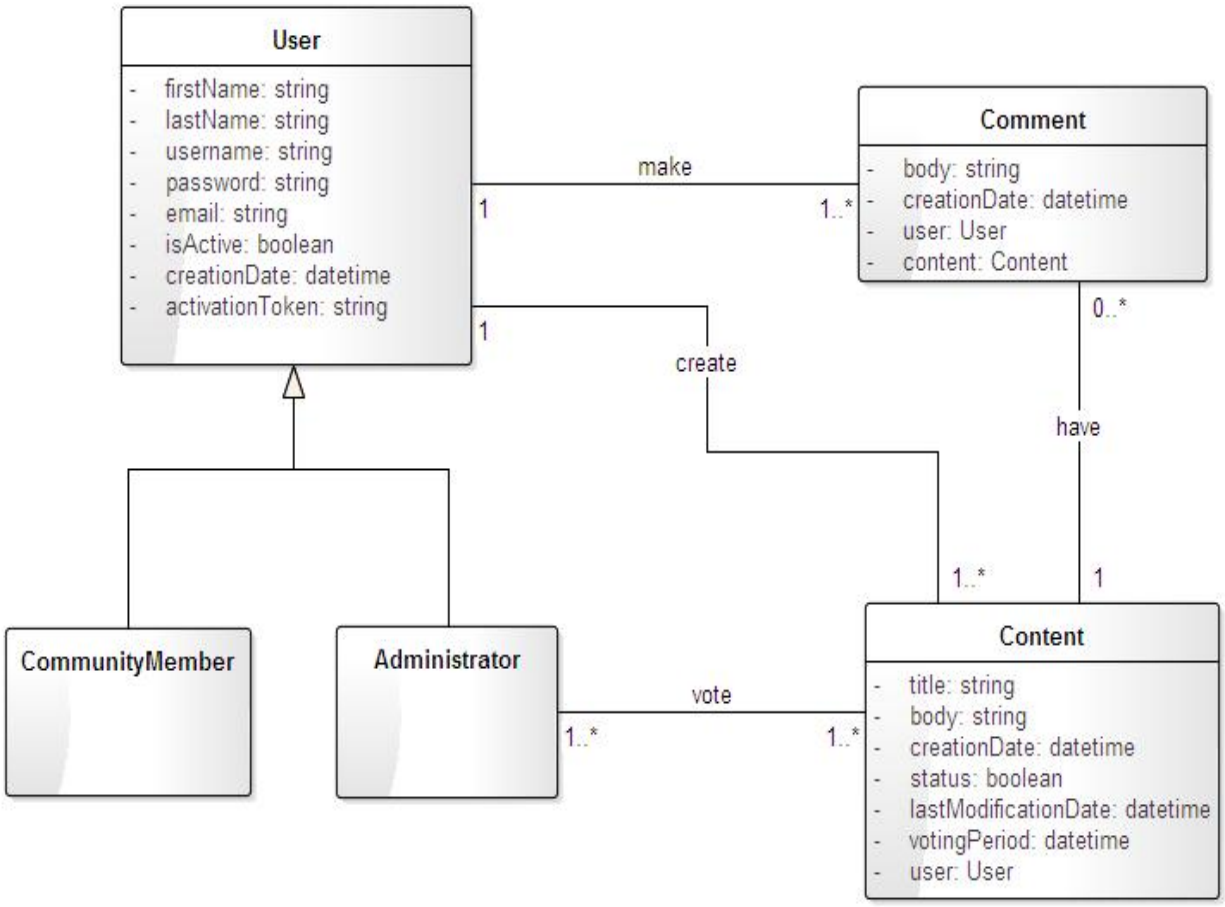

Figure 3.1: Conceptual Class Diagram of the Living Web Application

b. Object Diagram

Object diagrams show snapshot of the objects in a system at a point in time. They are often called an instance diagram because it shows intances rather than classes. Can be used to show example configuration of objects. Can be further used to simplify complex class diagrams (Fowler, 2004).

Below is an object diagram of the living web application. It represents a snapshot of the system at a particular moment.

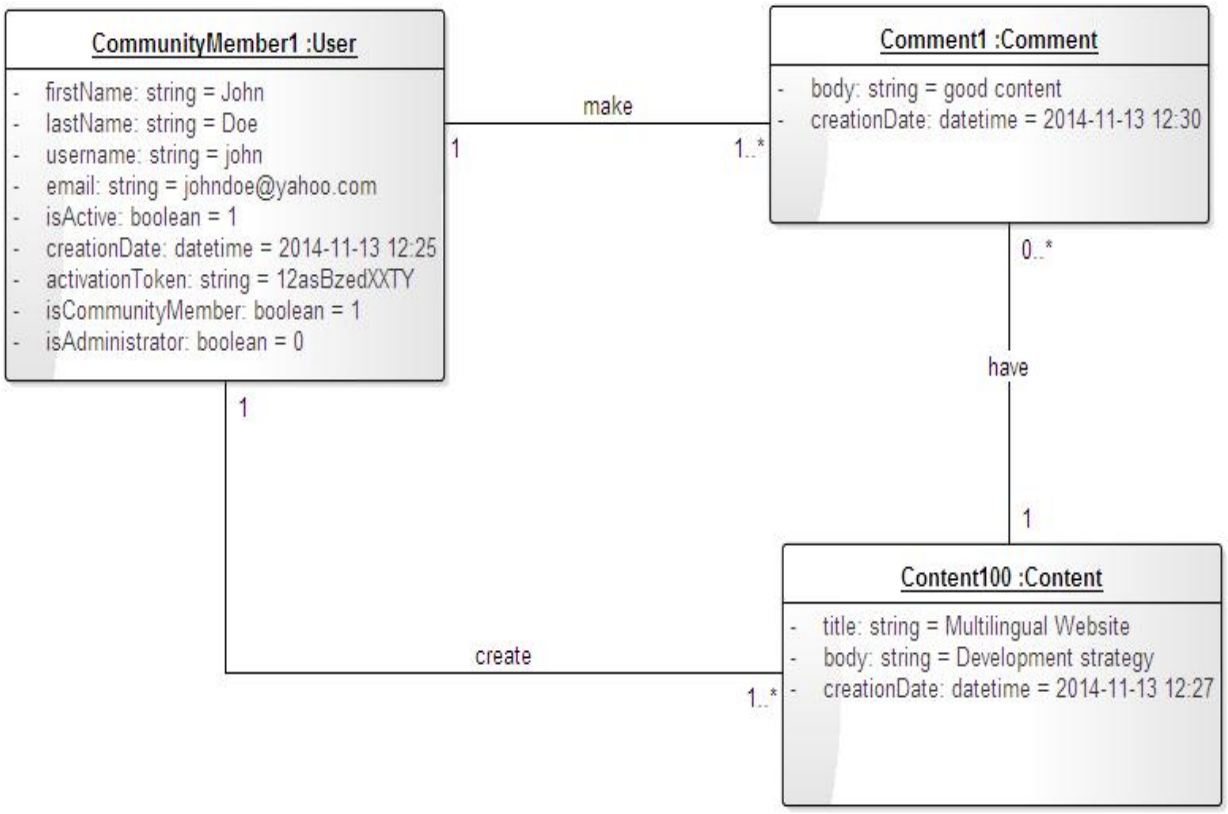

Figure 3.2: Object Diagram of the Living Web Application

c. State Machine Diagram

It is a technique used to describe the behavior of a system. It is very useful at describing the behaviour of an object across several use cases (Fowler, 2004). Below are state machine diagrams for the Administrator object and Community member object of the living web application. Figure 3.3 show the behaviour of the Administrator object across three different Use Cases which include: Sign Up, Login and Vote Content Use cases while Figure 3.4 show the behaviour of the Administrator Object across five different Use cases which include: Sign up, Create Content, Remove Content, Display Content and Select Content Use cases.

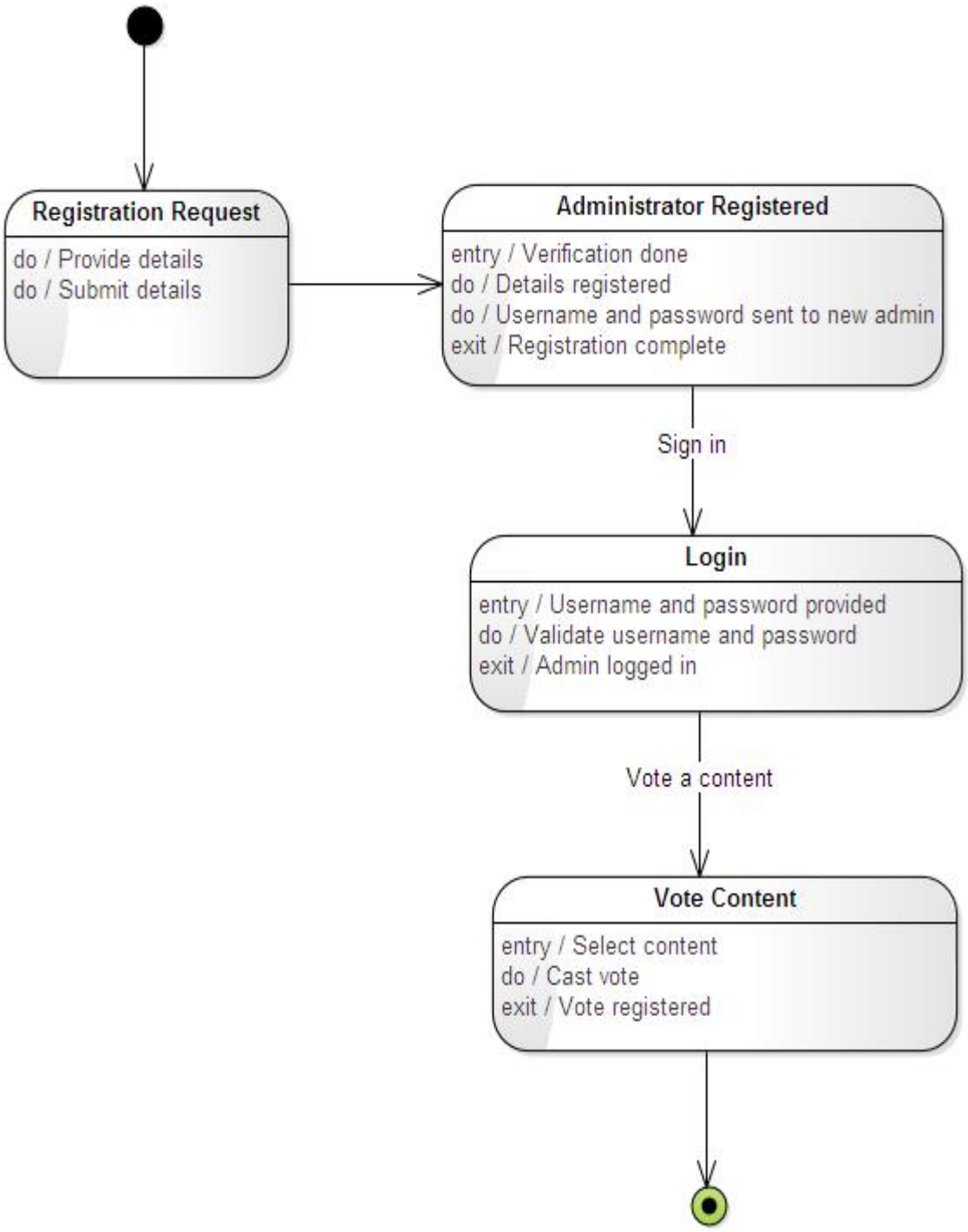

Figure 3.3: State Machine Diagram of the Administrator Object of the Living Web Application

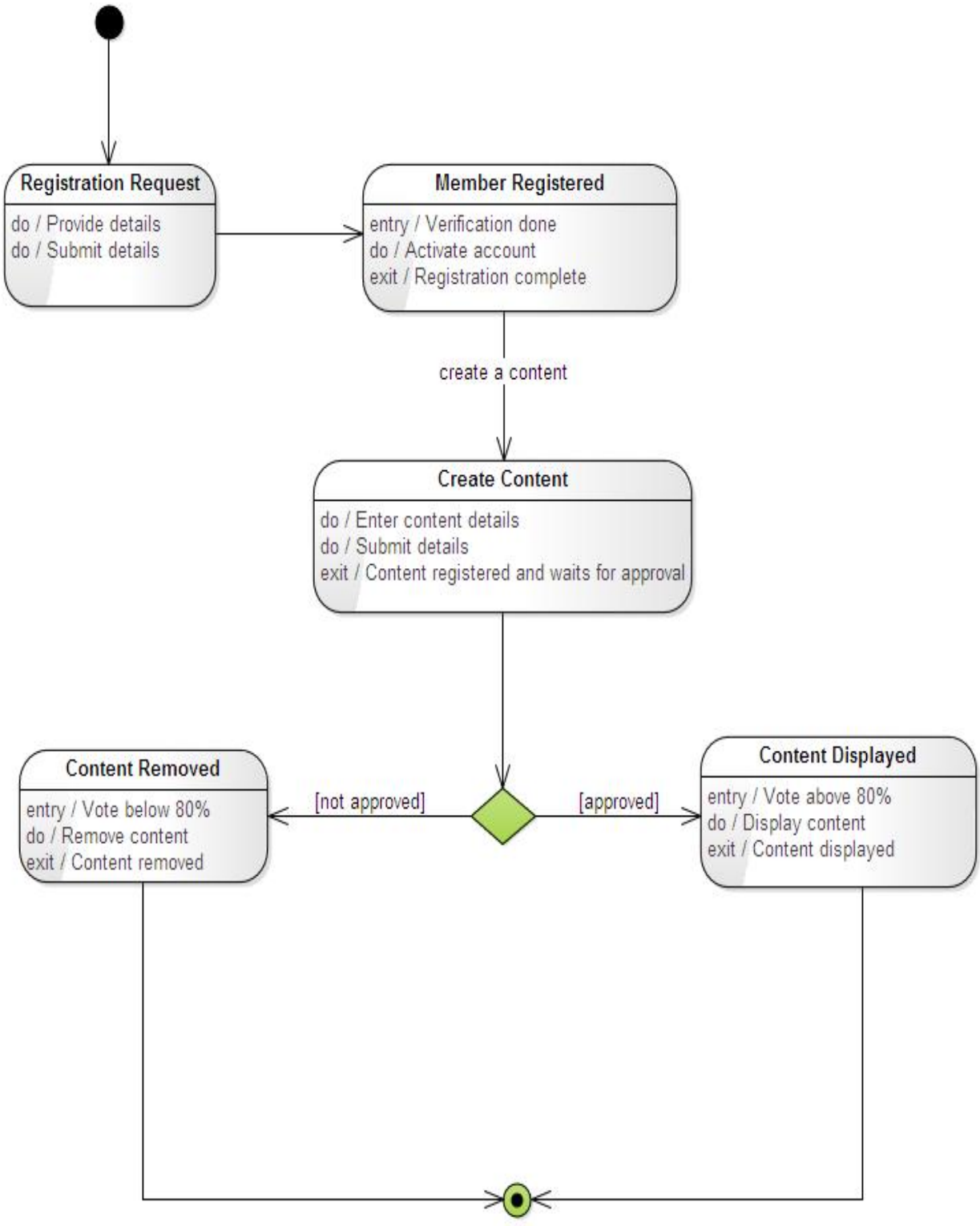

Figure 3.4: State Machine Diagram of the Community Member Object for the Living Web Application

3.2. Requirements

Requirements describe what the system should do. The following diagrams are used to model requirements: Use Case Diagram, Sequence Diagram and Activity Diagram.

a. Use Case Diagram

It is a technique for capturing the functional requirements of a system. Use Cases describe the typical interactions between the users of a system and the system it self, providing a narrative of how a system is used (Fowler, 2004). Shown below is a Use Case Diagram of the living web application.

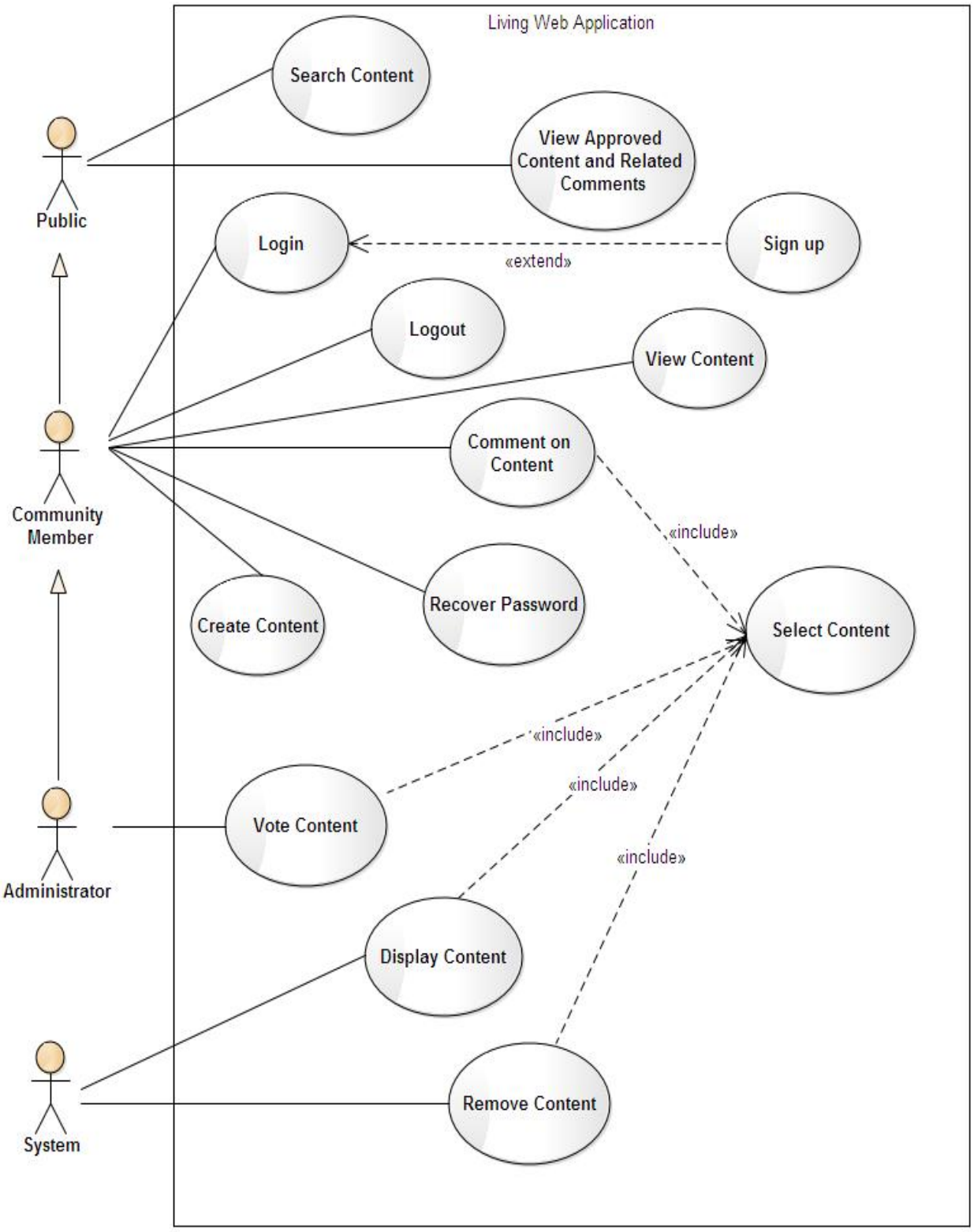

Figure 3.5: Use Case Diagram of the Living Web Application

b. Sequence Diagram

It describes how groups of objects collaborate in some behaviour. It captures the behavior of a single scenario. Shows objects and the messages that are passed between these objects within the Use Case. Sequence diagrams are very useful when you want to look at the behaviour of several objects within a single use case. Good at showing collaborations among objects (Fowler, 2004).

Shown below are sequence diagrams of the living web application. The following scenarios have been modeled.

- Community Member Sign up
- User Login
- Content Approval Process for a newly created content
- Password Recovery Process
- Content creation by a member
- View content by a public

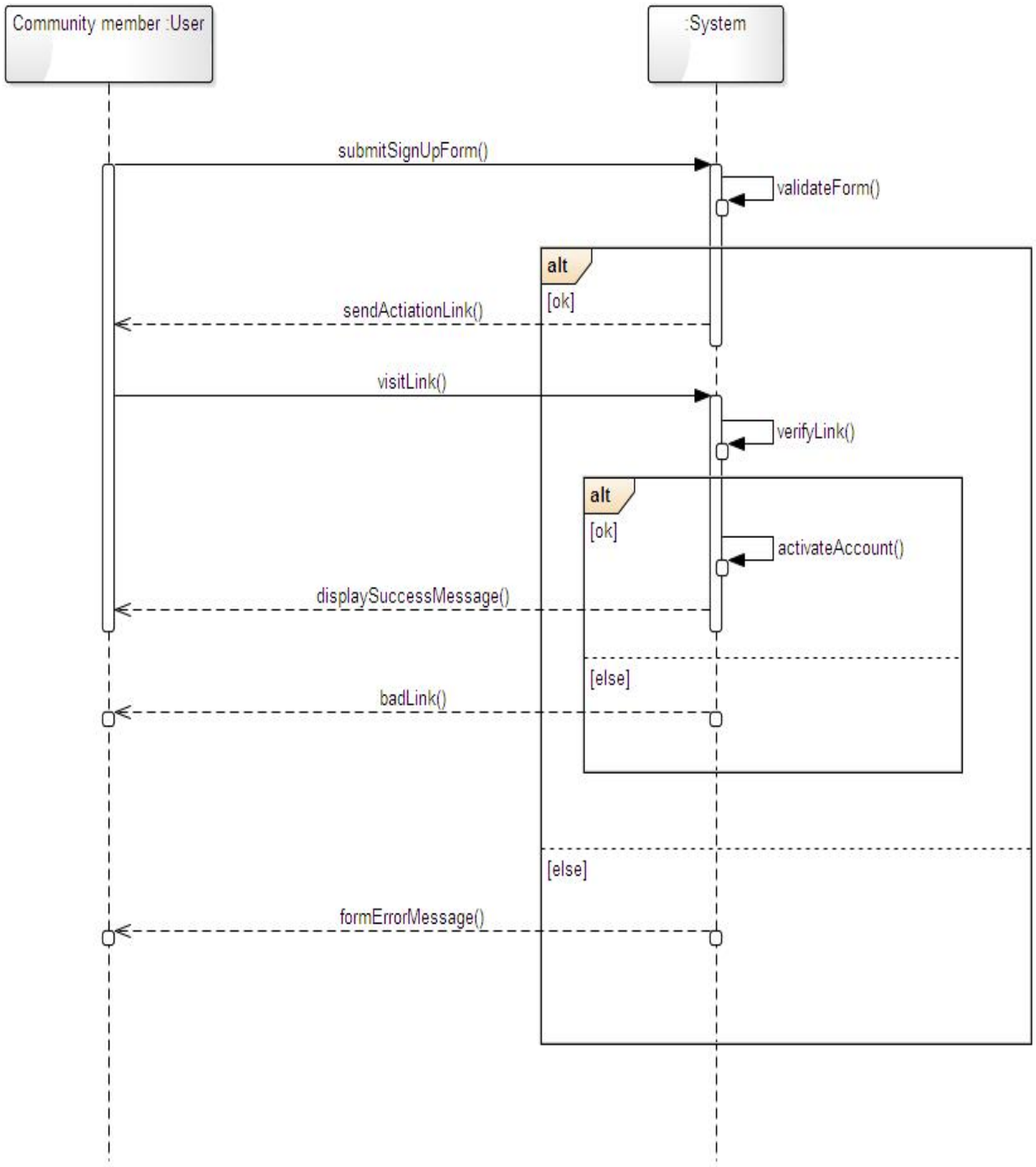

Figure 3.6: Sequence Diagram for Community Member Sign up

The sequence diagram above show the messages that are passed between the Community Member object and the system within the Sign Up use case.

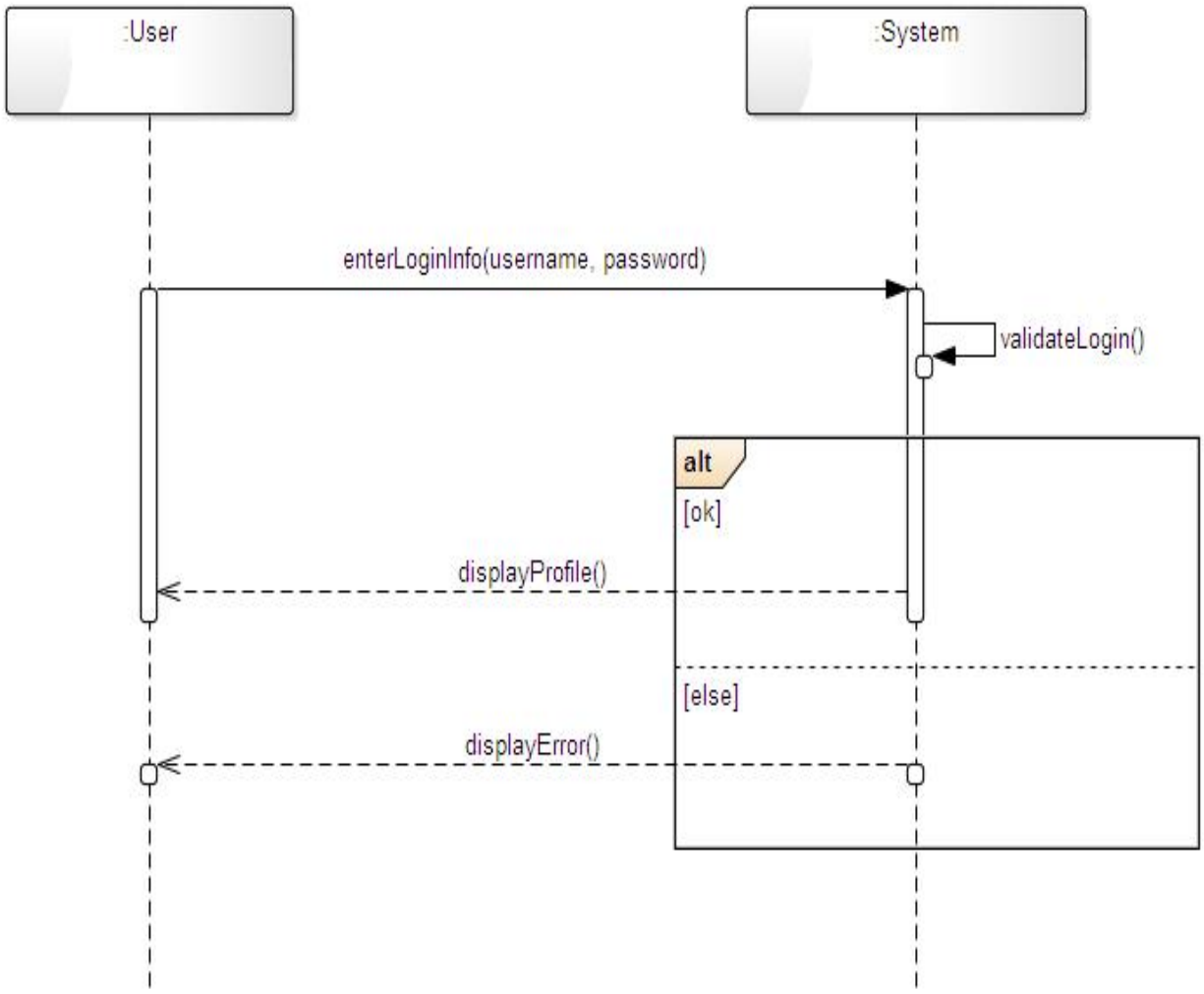

Figure 3.7: Sequence Diagram for User Login

The Sequence diagram above show the messages that are passed between the User object and the system within the Login use case. The User Object could be an administrator or a community member.

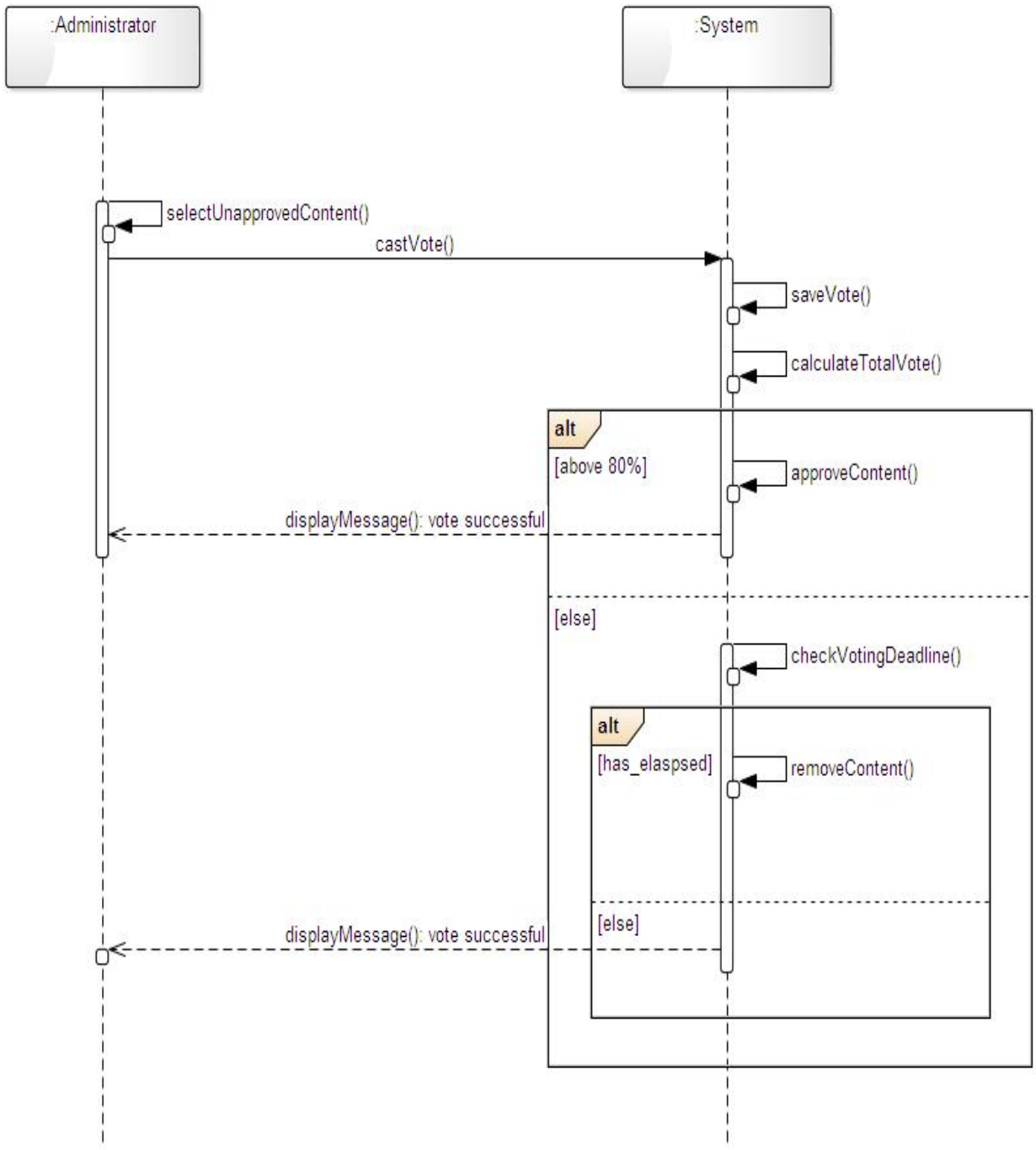

Figure 3.8: Sequence Diagram for Content Approval Process

The sequence diagram above shows the series of messages that are passed during the event when an administrator vote for a content that has been created right down to when this content is approved for votes greater then 80% or when it is removed for vote less then 80% after the voting deadline.

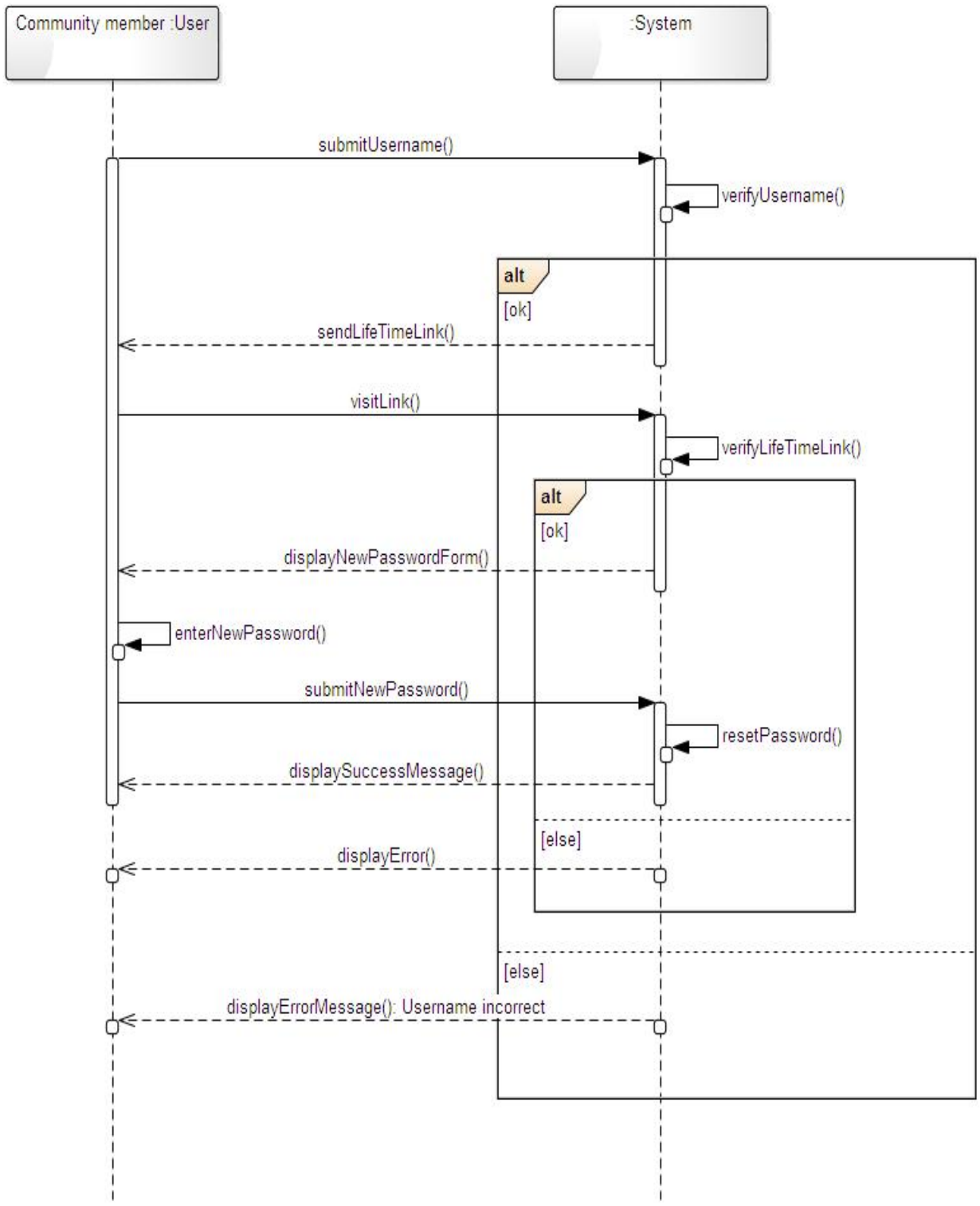

Figure 3.9: Sequence Diagram for Password Recovery Process

The sequence diagram above show the messages that are passed between the Community Member object and the system within the Recover Password use case.

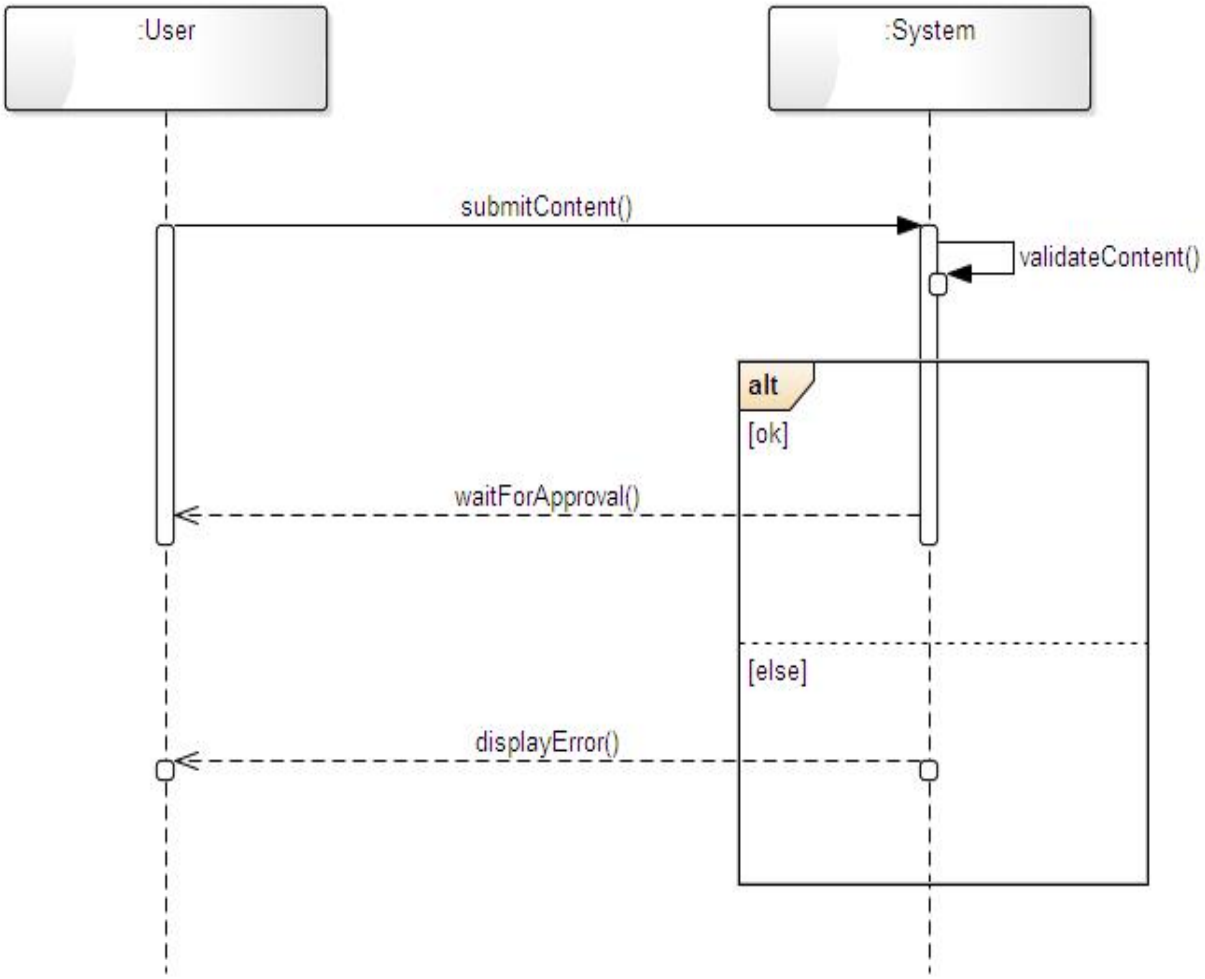

Figure 3.10: Sequence Diagram for Content Creation

The Sequence diagram above show the messages that are passed between the User object and the system within the Create Content use case. The User Object could be an administrator or a community member.

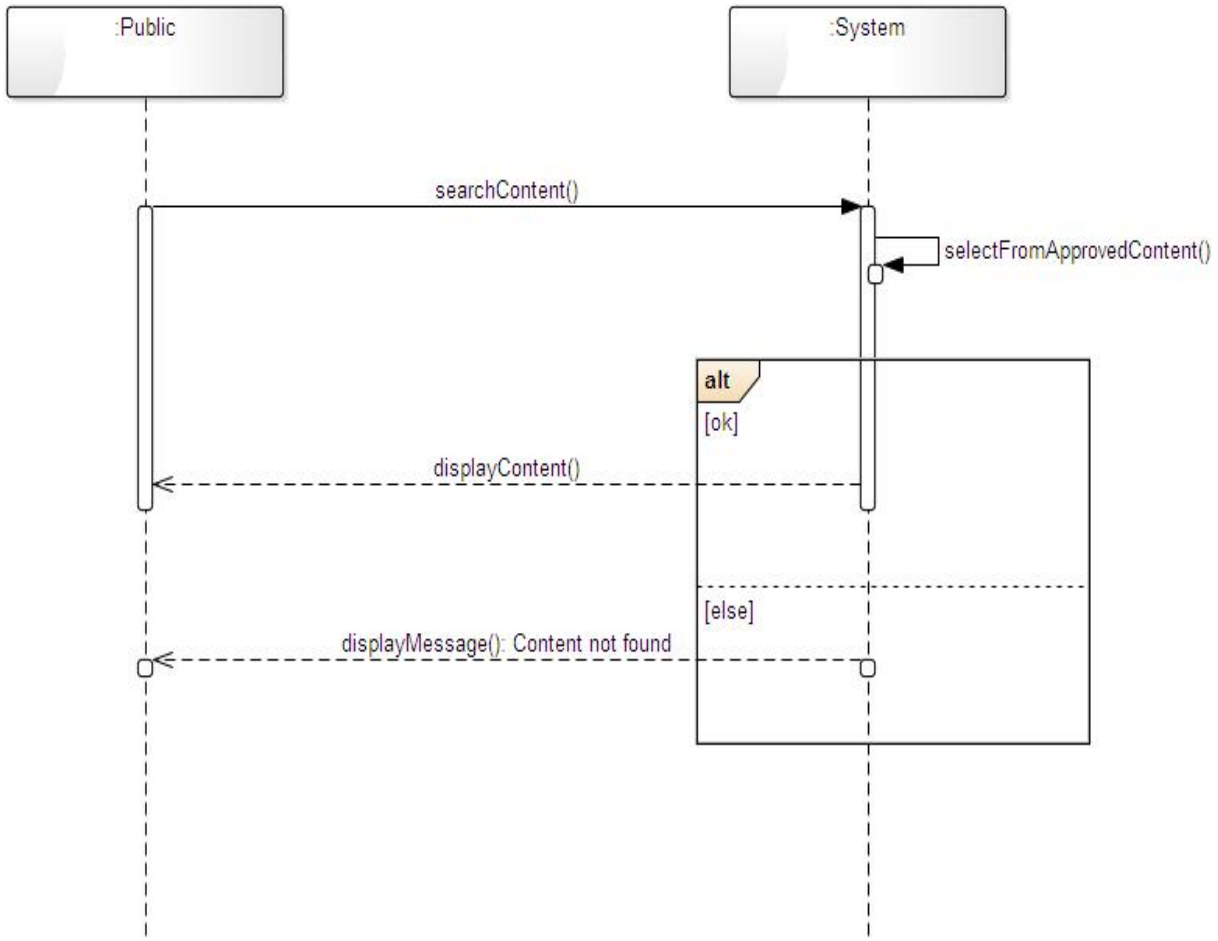

Figure 3.11: Sequence Diagram for View Content

The Sequence above show the messages that are passed between the Public object and the system within the View Content use case. Public refer to any one visiting the website.

c. Activity Diagram

It is a technique used to describe procedural logic, business process, and work flow. It supports parallel behaviour. It is useful for concurrent algorithms, in which independent threads can do things in parallel. Shown below are activity diagrams of the living web application. Activity diagrams for the following use cases are presented: Create Content, Vote Content, Recover Password and Sign up use cases.

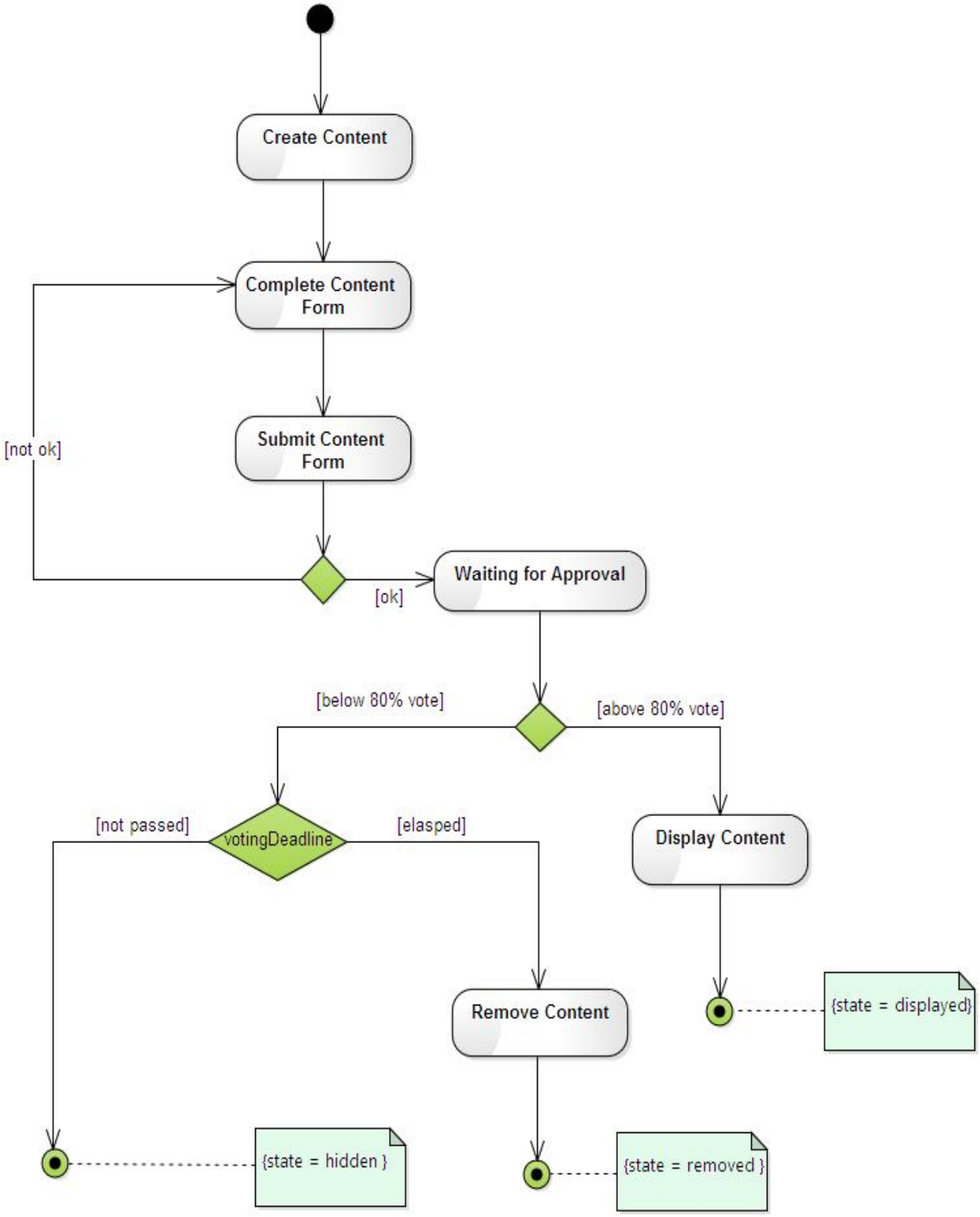

Figure 3.12: Activity Diagram showing the flow during Use Case "Create Content"

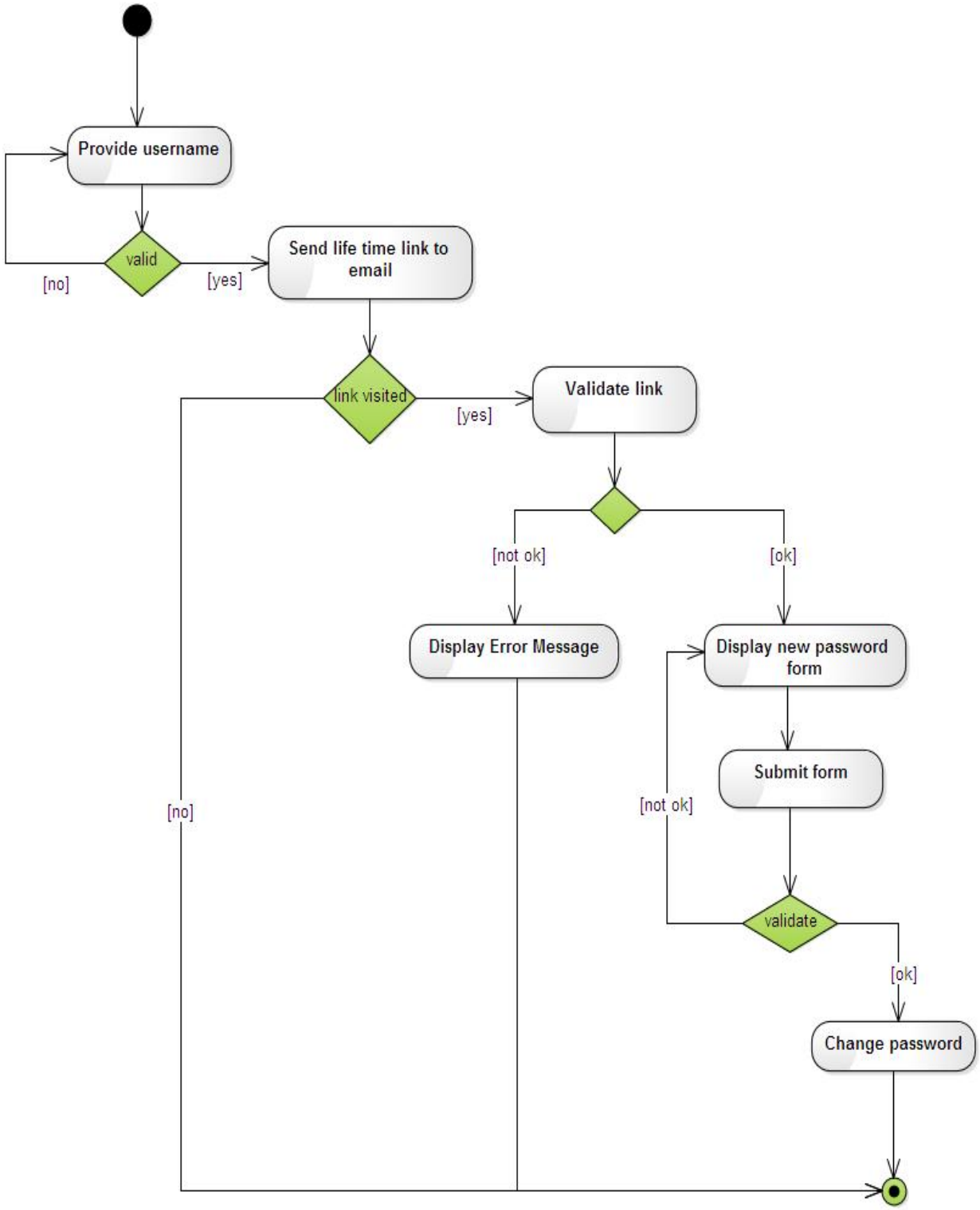

Figure 3.13: Activity Diagram showing the flow during Use Case "Recover Password"

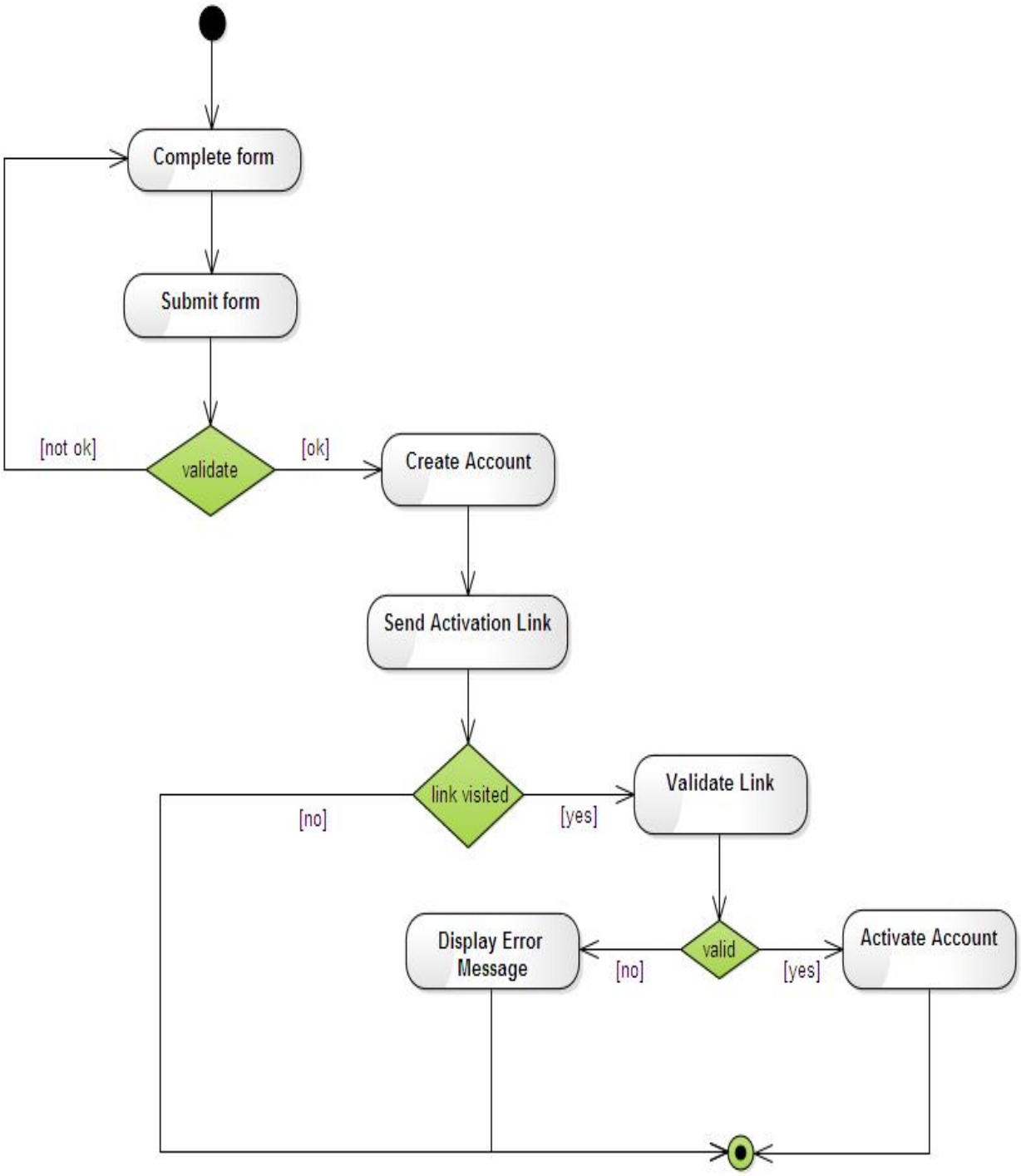

Figure 3.14: Activity Diagram showing the flow during Community member "Sign Up"

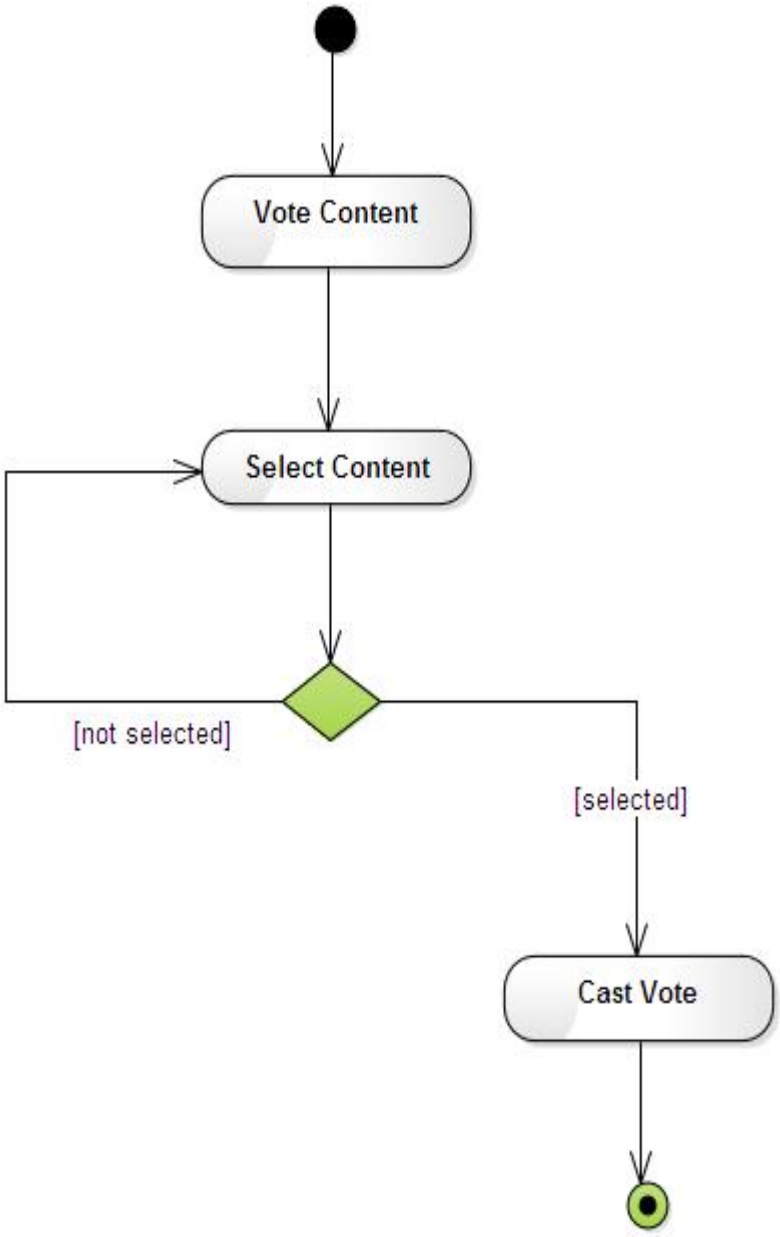

Figure 3.15: Activity Diagram showing the flow during Use Case "Vote Content"

3.3. Analysis and Design

Analysis and Design show how the system will be realized in the implementation phase. The following diagrams are used in this case: Design Class Digram, Component Diagram and Physical Data model. These diagrams answer the question as how a living web application can be implemented.

a. Design Class Diagram (DCD)

DCD illustrate the specification of software classes (Larman). Design Classes are easily derived from conceptual classes. It shows the definition of software classes rather than real world concepts. Shown below is the design class diagram of the living web application.

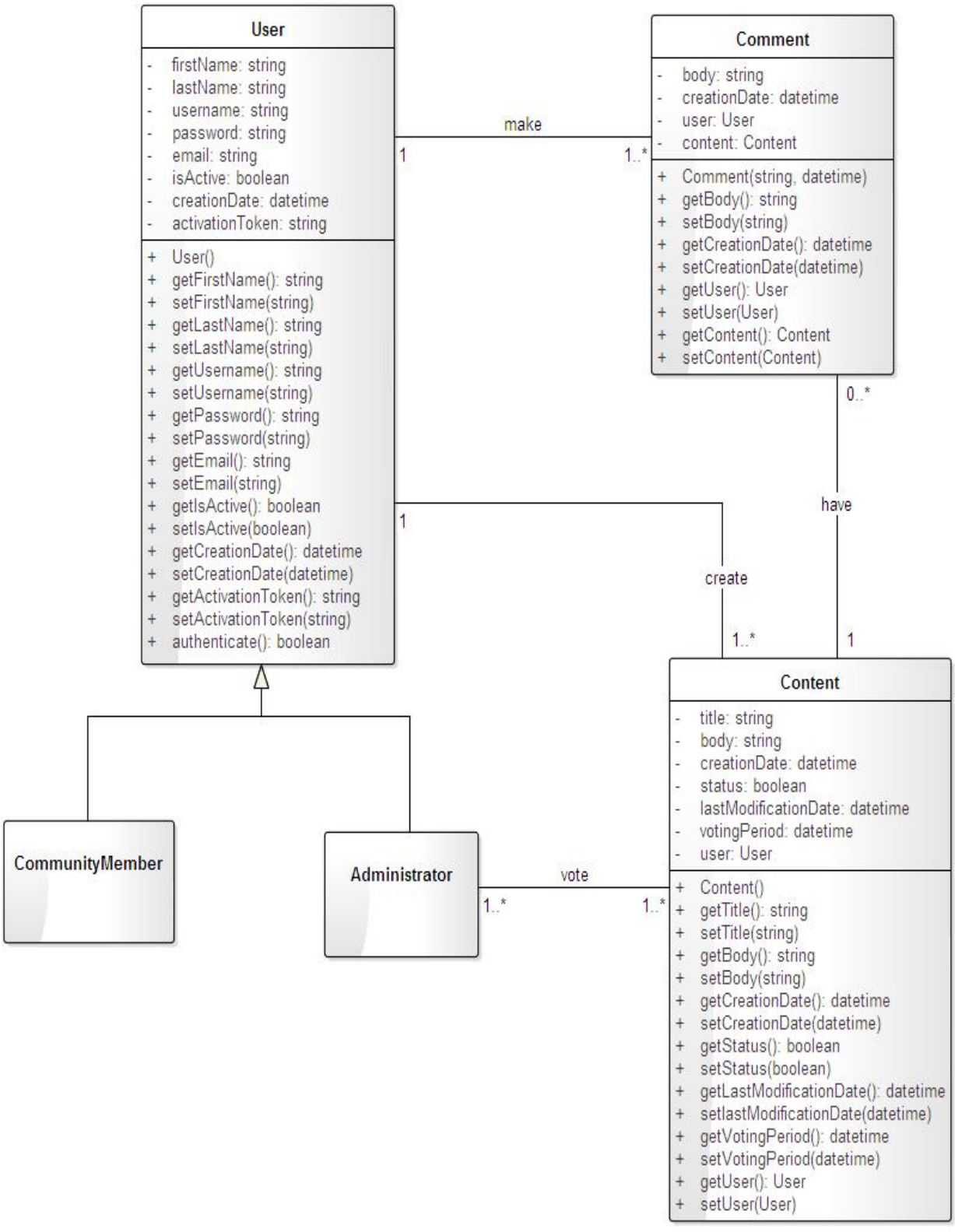

Figure 3.16: Design Class Diagram of the Living Web Application

b. Component Diagram

Component diagrams addresses how customers relate to software. Represent pieces that are independently purchasable and upgradeable (Fowler, 2004). The components represent cleanly grouped and encapsulated elements from the logical architecture. Shown below is the component diagram of the living web application. It shows how different components together provide the persistent aspect of the system. Each component has a set of responsibilities which together contibute to the functioning of the entire system. The Components User, Content and Comment all together provide the persistent layer of the application. The Living Web Application component require all these three components for it's complete functionality to be fulfilled.

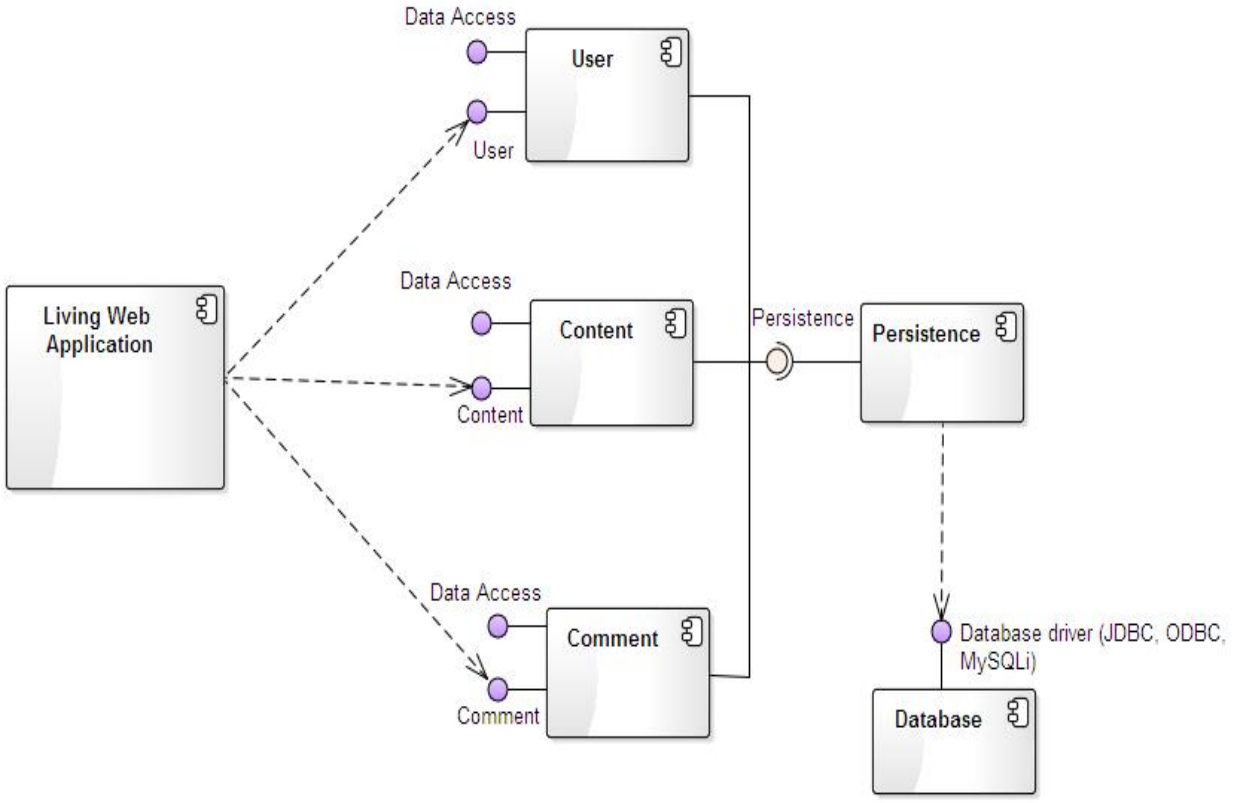

Figure 3.17: Component Diagram of the Living Web Application

c. Physical Data Model

Data modeling describes how the information domain of the system is transformed into data structures. It provides a description of how the major entities are stored, processed and organized. Object relational mapping is used to show how objects and their relationships are stored in relational databases. Only the data-oriented aspects of the design classes are considered. Shown below is the physical data model which is drawn using UML notation.

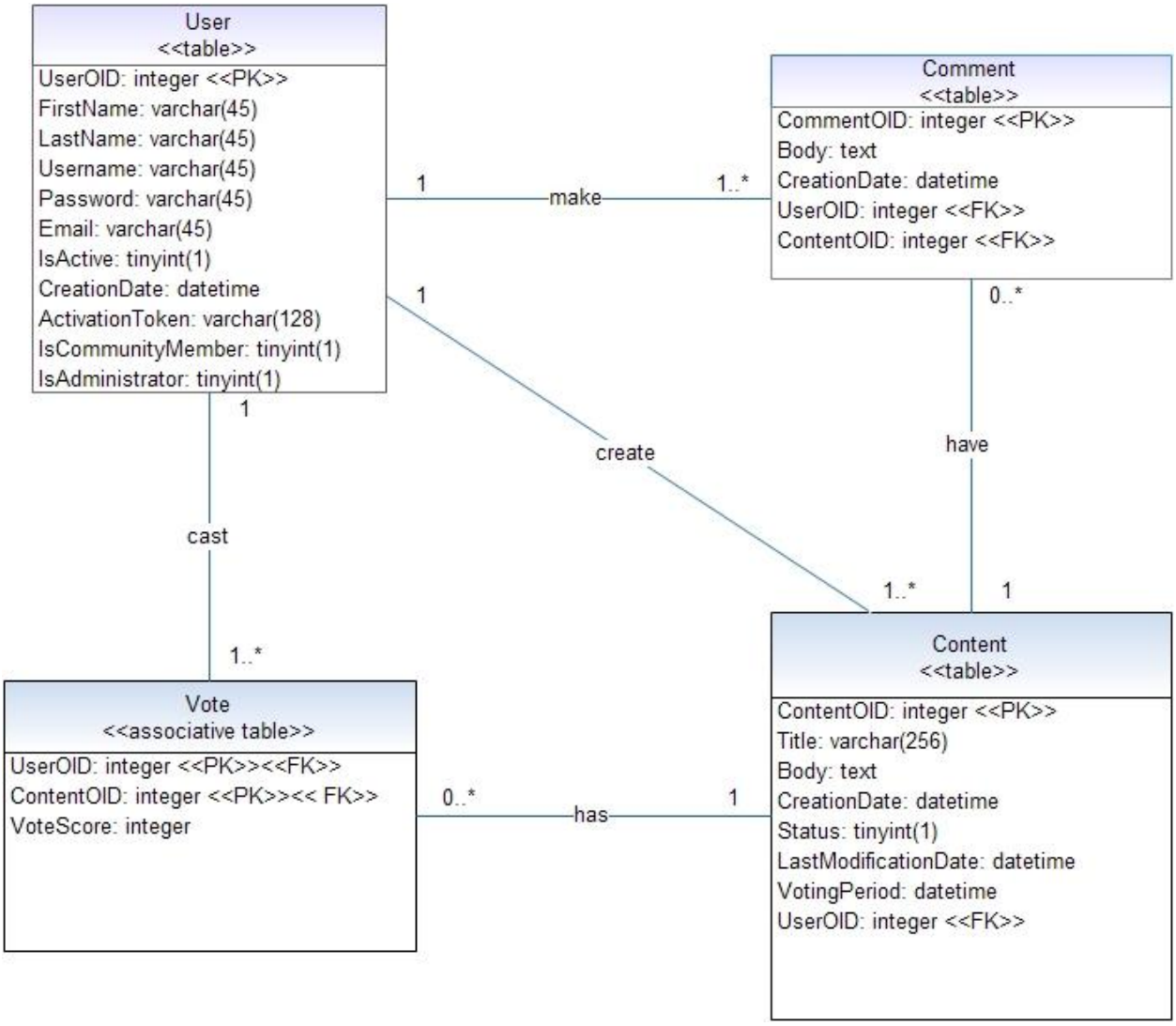

Figure 3.18: Physical Data Model (UML notation)

4. Conclusion

This paper has made used of techniques in Unified Modeling Language (UML), Rational Unified Process (RUP) and Object Relational Mapping (ORM) to systematically present the concept of a living web application, its logical functioning and a propose implementation. Different UML diagrams have been used across different workflows in RUP to present the concept in a coherent manner. The study presented in this paper is limited as not every aspect of a living web application has been analyzed thus there is room for further analysis, but the paper provides valuable insights for understanding the concept of the abovementioned study topic. The authors make no claim that the models provided in this study are the best and should be used as a standard but acknowledge the fact that a system can be analyzed and view differently by different analyst. So this study is only a possible solution to a concept. The different UML diagrams are drawn using Enterprise Architect version 11.0.1106 while the physical Data model is drawn using Edraw Max version 7.7.

5. References

Adegboyega Ojo, E. E. (n.d.). Object-Oriented Analysis and Design with UML.

Agile Unified Process. (n.d.). Retrieved from http://en.wikipedia.org/wiki/Agile_Unified_Process

Ambler, S. (2004). *The Object Primer 3rd Edition Agile Model Driven Development with UML 2.* Cambridge University Press.

Fowler, M. (2004). *UML Distilled.* Addison-Wesley.

Larman, C. *Applying UML and Patterns.*

RUP - Rational Unified Process. (n.d.). Retrieved from http://www.webopedia.com/TERM/R/RUP.html

Wikipedia. (n.d.). *Model-View-Controller*. Retrieved from http://en.wikipedia.org/wiki/Model-view-controller

Wikipedia. (n.d.). *UML*. Retrieved from http://en.wikipedia.org/wiki/Unified_Modeling_Language