<Image Gallery Software>

Use-Case-Realization Specification: <Register>

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <1.0> | <details> | Nguyen Kiem Hoa |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 References 4

1.5 Overview 4

2. Use-Case Specification 4

3. Interaction Diagrams 5

4. Class Diagrams 6

5. Derived Requirements 7

Use-Case-Realization Specification: <Register>

# Introduction

## Purpose

This document describes how the Register Use-Case is realized within the design model, in terms of collaborating objects.

## Scope

This document applies to the Image Gallery Software which will be developed Group 5C-18.

## Definitions, Acronyms, and Abbreviations

User – a person who use the software.

## References

None.

## Overview

In the following section, Use-Case Realization Specification of the Register Use-Case of the Image Gallery Software is provided in detail. The first section is a textual description of the Use-Case specification. The following section contains diagrams (sequence and collaboration diagrams) describing how the use case is realized in terms of collaborating objects. The third section includes class diagrams with relationships that participate in the realization of the use case. The last section is an analysis of all requirements, such as non-functional requirements, on the use-case realization that are not considered in the design model, but that need to be taken care of during implementation.

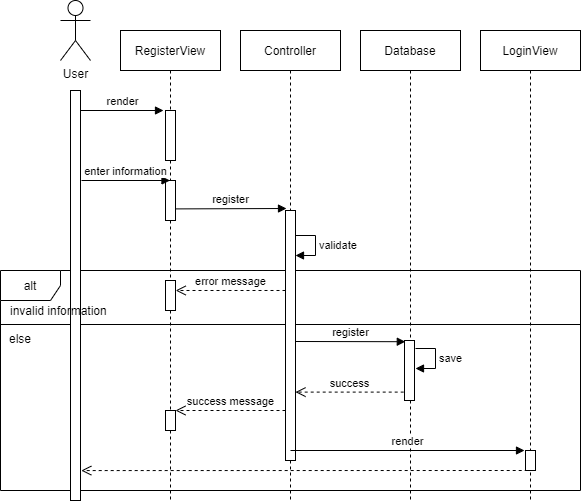
# Use-Case Specification



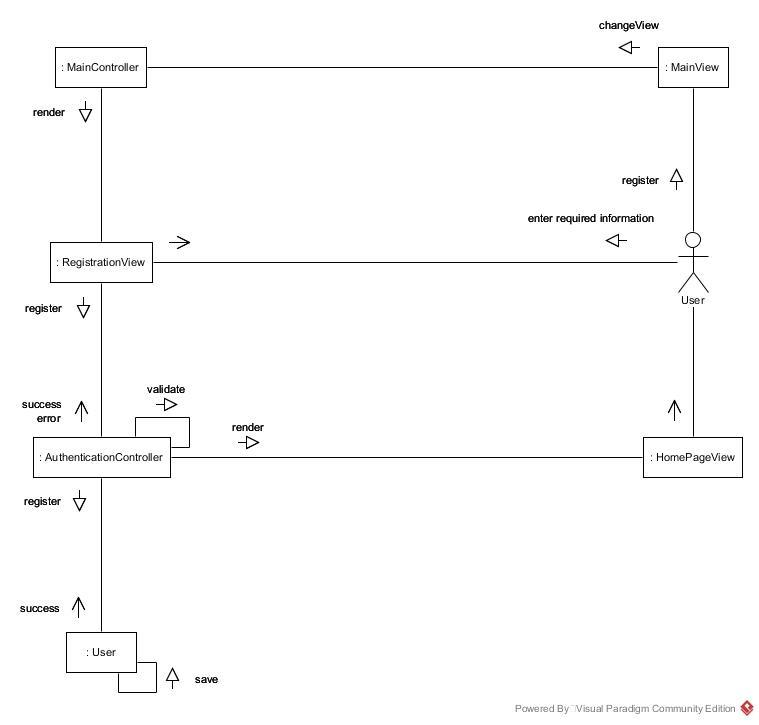
|  |  |
| --- | --- |
| **Name** | Register |
| **Brief Description** | A user creates an account |
| **Actor** | User |
| **Flow of Events** | |
| **Basic Flow** | |
| 1. The user select registration option while try to login the system. 2. The system prompts user for registration information: username, password, etc. 3. The user enters their information. 4. The system validates user information. 5. The system creates a new user base on entered information. 6. The system saves the new user to database. | |
| **Alterante Flows** | |
| **Title** | **Description** |
| Invalid information | 1. User enter invalis information 2. System displays message to correct invalid information. |
| **Pre-Conditions** | |
| Not login | |
| **Post-Conditions** | |
| **Title** | **Description** |
| Success | Success message is prompted and user enter login screen. |
| Failure | The user is unable to register because invalid information and an error is prompted. |
| **Extension Points** | |
| None | |

# Interaction Diagrams

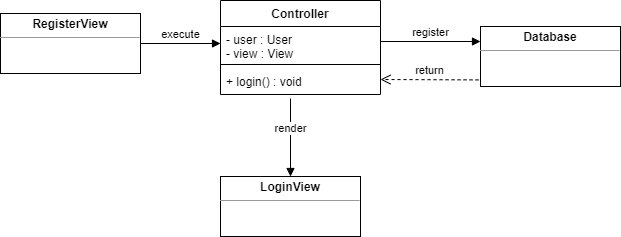
**Sequence Diagram:**



**Collaboration Diagram:**



# Class Diagrams



# Derived Requirements

None.