

VOLUNTARY NONE REMUNERATED BLOOD DONATION

Architecture Design

**Project Code: VNBD**

**Document Code: VNBD\_ Architecture Design\_v1.0\_EN**

**Ha Noi, 25/09/2013**

Record of change

\*A - Added M - Modified D - Deleted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Effective Date | Changed Items | A\* M, D | Change Description | New Version |
| 25/09/2013 |  | A | Add new | v0.1 |
| 20/10/2013 | Deployment View | M | Modify deployment view | v0.9 |
| 26/10/2013 | Sequence Diagram | M | Modify all sequence diagram | v1.0 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Table 1: Record of change

SIGNATURE PAGE

ORIGINATOR: Le Huy Hoang 25/09/2013

Developer

REVIEWERS: Ngo Tran Linh 08/10/2013

PM

APPROVAL: Bui Dinh Chien 27/10/2013

Supervisor

TABLE OF CONTENTS

[1 Introduction 5](#_Toc370650738)

[1.1 Purpose 5](#_Toc370650739)

[1.2 Scope 5](#_Toc370650740)

[1.3 Definitions, Acronyms and Abbreviations 5](#_Toc370650741)

[1.4 References 5](#_Toc370650742)

[1.5 Overview 5](#_Toc370650743)

[2 Architectural Representation 6](#_Toc370650744)

[3 Architectural Goals and Constraints 7](#_Toc370650745)

[3.1 Technical platform 7](#_Toc370650746)

[4 Use-Case View 7](#_Toc370650747)

[4.1 Use-Case Realizations 8](#_Toc370650748)

[5 Logical View 10](#_Toc370650749)

[5.1 Overview 11](#_Toc370650750)

[5.2 Architecturally Significant Design Packages 11](#_Toc370650751)

[6 Process View 12](#_Toc370650752)

[7 Implementation View 13](#_Toc370650753)

[7.1 Overview 13](#_Toc370650754)

[7.2 Layers 28](#_Toc370650755)

[8 Deployment view 28](#_Toc370650756)

[9 Size and Performance 28](#_Toc370650757)

[10 Quality 28](#_Toc370650758)

[11 Other Considerations 29](#_Toc370650759)

# Introduction

## Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions that have been made on the system.

## Scope

The scope of this document is to depict the architecture of the Voluntary non-remunerated blood donation website created by VNBD capstone project team.

## Definitions, Acronyms and Abbreviations

|  |  |  |
| --- | --- | --- |
| Acronym | Definition | Note |
| VNBD | Voluntary non-remunerated blood donation |  |
| MVC | Model view control |  |
| IDE | Integrated development environment |  |
| Q&A | Question and answer |  |
| GUI | Graphic user interface |  |

Table 2: Definitions

## References

* VNBD\_ Software Requirements Specification\_v1.0\_EN.docx
* VNBD\_Data Design\_v0.9.9\_EN.docx
* Sample Design at Introduction to software engineering (I2SE) course in cms.fpt.edu.vn
* Software Architecture Design Illuminated Book
* <http://en.wikipedia.org/wiki/Model-view-controller>
* <http://en.wikipedia.org/wiki/Model–view–controller>
* <http://fit.hcmup.edu.vn/~hienlth/SE/Lecture_Notes/GioiThieuRUP.pdf>

## Overview

The Software Architecture Document contains the following subsections:

* Section 1: Provide an overview of entire Software Architecture Document.
* Section 2: Architectural Presentation.
* Section 3: Architectural goals and constraints.
* Section 4: Use-case view.
* Section 5: Logical view.
* Section 6: Process view.
* Section 7: Implementation view.
* Section 8: Deployment view.
* Section 10: Size and performance.
* Section 11: Quality.

# Architectural Representation

* This section describes what architecture design is for the current system, and how it is represented. Of the **Use-Case**, **Logical**, **Process**, **Deployment**, and **Implementation Views**, it enumerates the views that are necessary and explains what types of model elements it contains for each view.
* This section details the architecture using views defined in “4+1 architecture view model”.

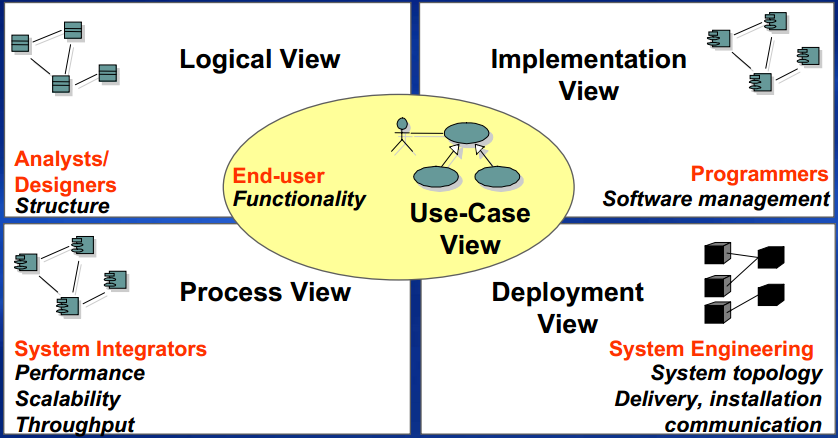


Figure 1: “4+1 view model”

|  |  |  |  |
| --- | --- | --- | --- |
|  | Audience | Area | Related artifacts |
| Use-case view | All stakeholders including end-users. | Describes the set of scenarios and use cases that represent some significant functionality of the system. | Use-case model, use-case document |
| Logical view | Designers | Functional requirements: describes the design’s object model. Also describes the most important use-case realization. | Design model |
| Process view | Integrators | Non-functional requirements: Describes system’s decomposition into lightweight processes (single threads of control) and heavyweight processes (groupings of lightweight processes). | None |
| Implement view | Programmers | Software components: describes the layers and subsystems of application. | Implementation model, component |
| Deployment view | Deployment managers | Describes the mapping of the software onto the hardware. | Deployment model |

Table 3: “4+1” view model explanation

# Architectural Goals and Constraints

This section describes the software requirements and objectives that have some significant impact on the architecture.

## Technical platform

* IDE: Dreamweaver 8.0, Notepad++.
* Framework: Zen framework.
* Web server: XAMPP 1.8.2.
* PHP 5.4.

# Use-Case View

The following user is the list of actor that will interact with the website.

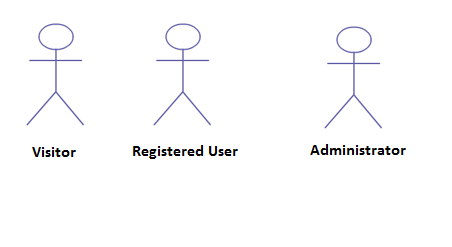


Figure 2: Actors of website

## Use-Case Realizations

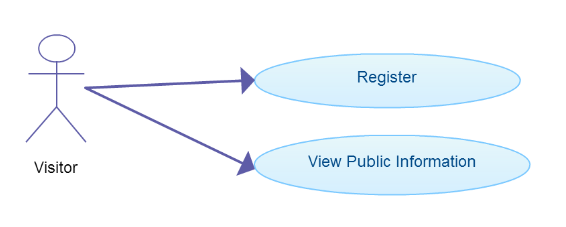


Figure 3: Visitor use-case

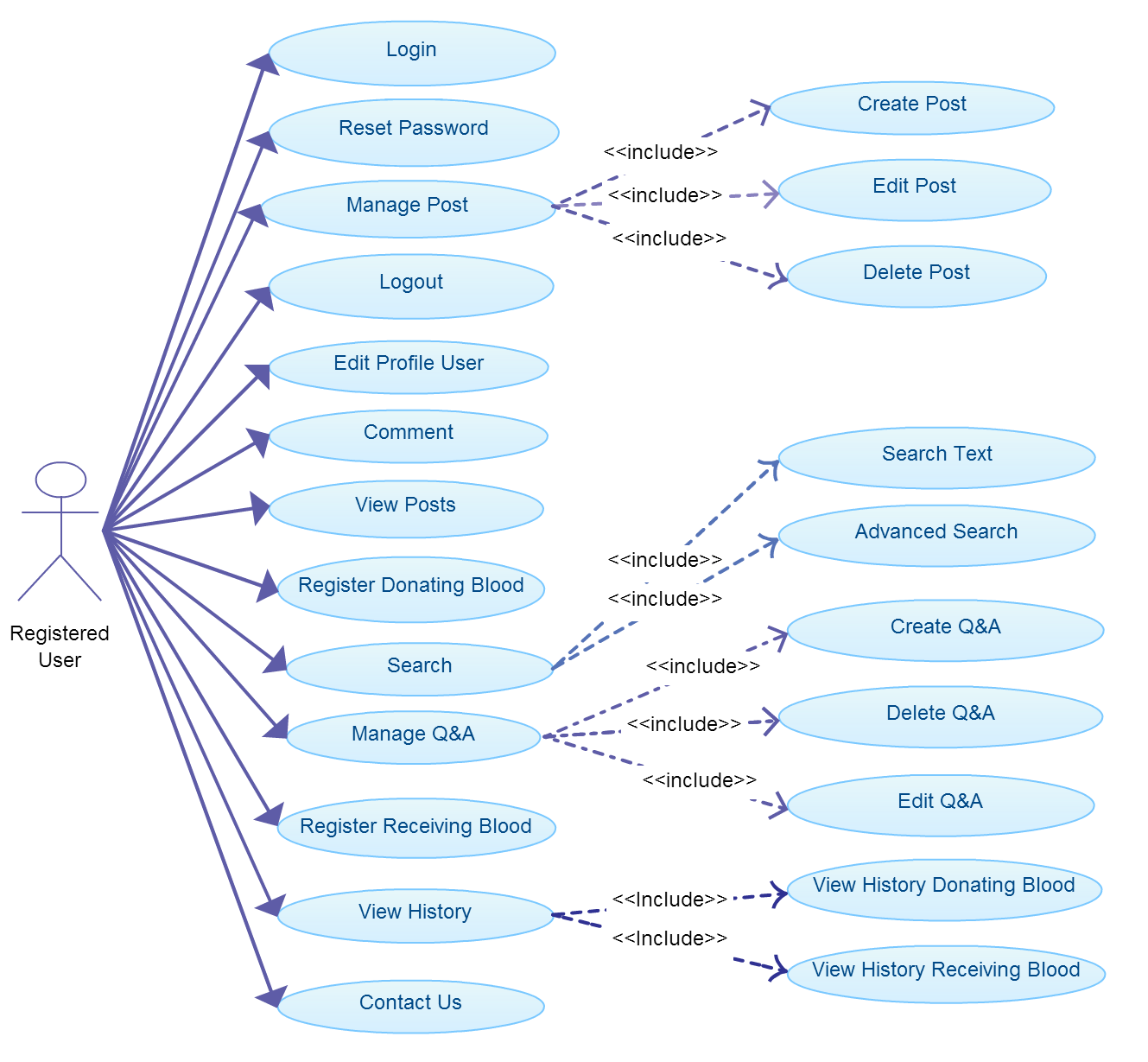


Figure 4: Registered use-case

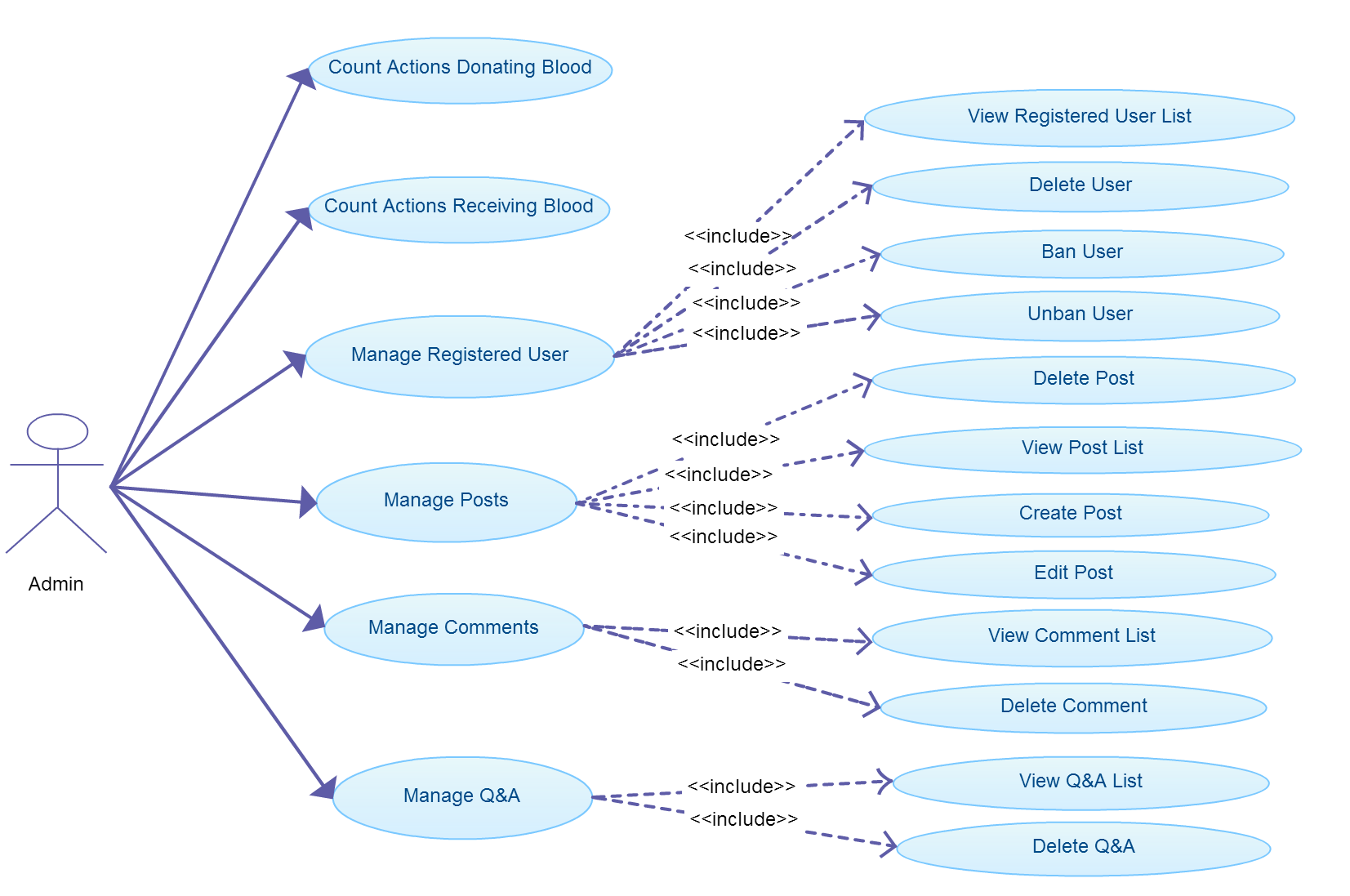


Figure 5: Admin use-case

# Logical View

This website based on Model view controller model.

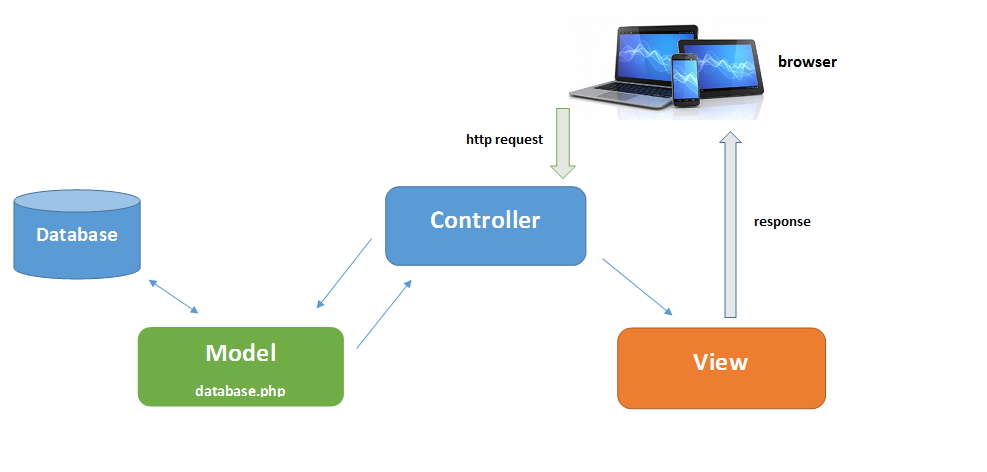


Figure 6: MVC model

## Overview

* Controller contain the interface between
* Associated models
* Associated views
* The input devices (e.g., keyboard, pointing device, time).
* Send commands to the model to update the model's state.
* Model is:
  + the domain-specific software simulation
  + Or implementation of the application's central structure.
* View deal with everything graphical
* Requests data from their model
* Display the data

## Architecturally Significant Design Packages

* Software Architecture-Layer Diagram

Presentation

Libraries (js, css, images)

Model

Exception

Logging

Common

Database Access

Search

Donating and Receiving blood

User management

Admin management

Q&A

Business

Data Access

Database

GUI

Comment management

News management

VNBD (Mysql)

Figure 7: System architect overview

# Process View



Figure 8: Process view

* Main screen: Process start when visitor connect to the website. In this process, user can view posts, images…Process end when visitor click to login, register or close website.
* Login: When process starts website redirect to login form. After user filling username, password and click login, system will search data in database. If success website will redirect to registered user’s information. Process end when user logout or click donating blood/receiving blood’s function.
* Register donating blood: Pre-condition: User logged in.
* Process starts when user click to register donating blood, end when user submit form register donating blood, click to other functions or close website.
* Register receiving blood: Pre-condition: User logged in.
* Process starts when user click to register receiving blood, end when user submit form register receiving blood, click to other functions or close website.

# Implementation View

## Overview

* Component diagram

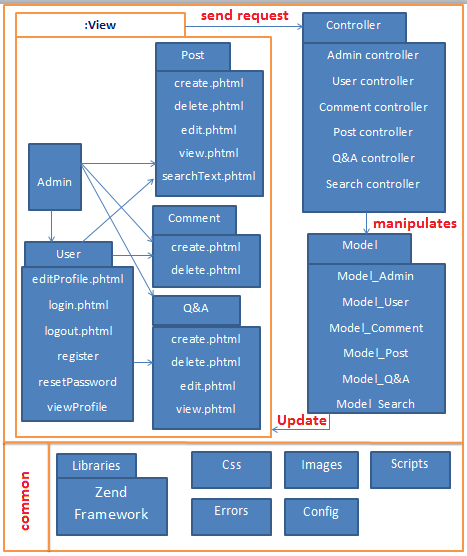


Figure 9: Component Diagram

* Sequence diagram



Figure 10: Register new account

* View public information



Figure 11: View public information

* Login



Figure 12: Login

* Reset password



Figure 13: Reset password

* Logout



Figure 14: Logout

* Edit profile

Figure 15: Edit profile

* Comment



Figure 16: Comment

* View posts



Figure 17: View posts

* Register donating blood.



Figure 18: Register donating blood

* Q&A management



Figure 19: Q&A management

* Registered user comment



Figure 20: Registered user comment

* Search Text

Figure 21: Search text

* Advanced Search

Figure 22: Advanced search

* Register receiving blood



Figure 23: Register receiving blood

* User manage their posts

Figure 24: User manage their posts

* Administrator manage user (view list, delete, ban, unban, delete user)

Figure 25: Administrator manage user

* Contact us

Figure 26: Contact us

* Administrators manage user’s comment

Figure 27: Administrator manage user’s comment

* Administrators manage Q&A (View list, delete)

Figure 28: Administrator manage Q&A

* Class diagram
* User package

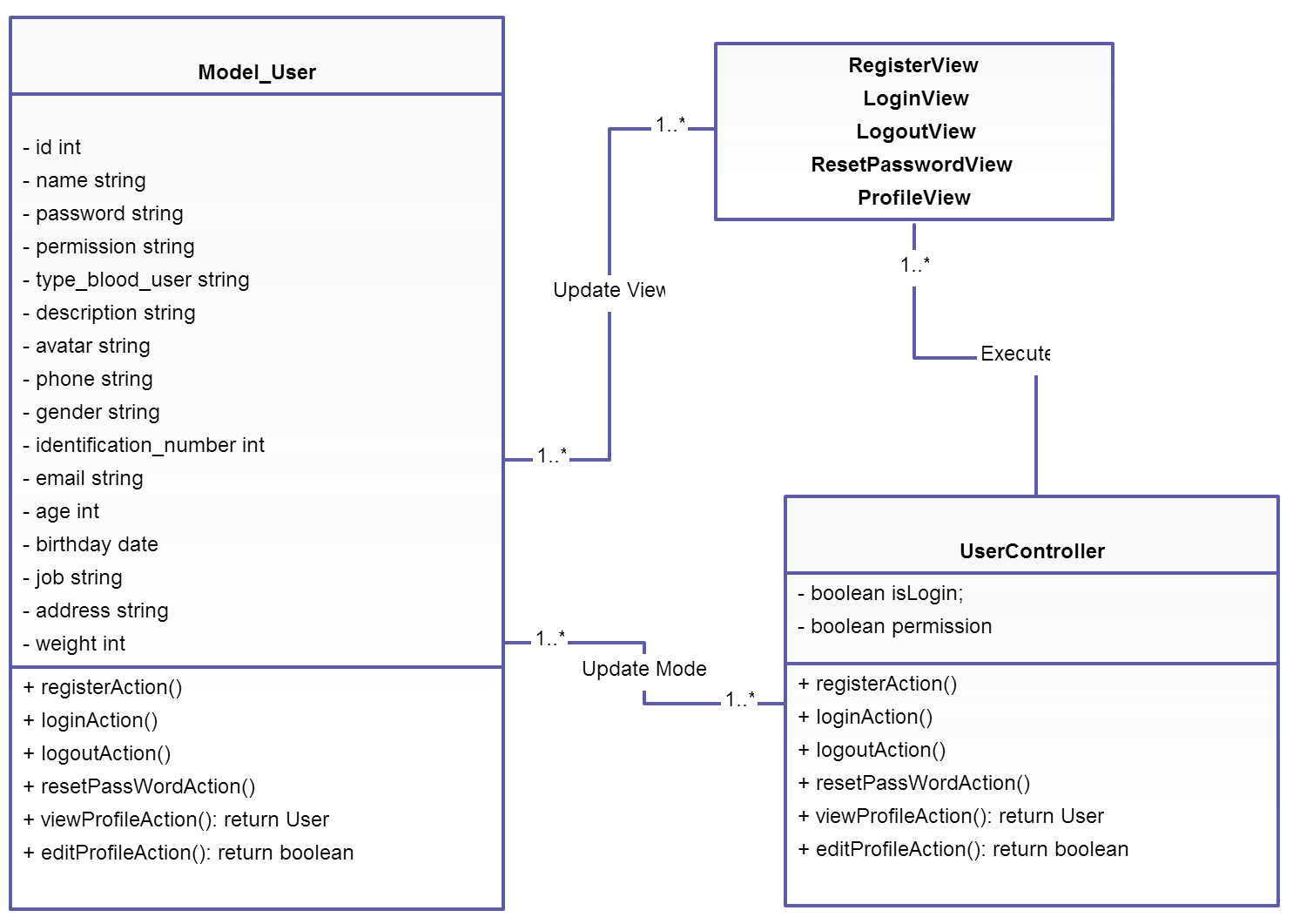


Figure 29: User package

* Post package

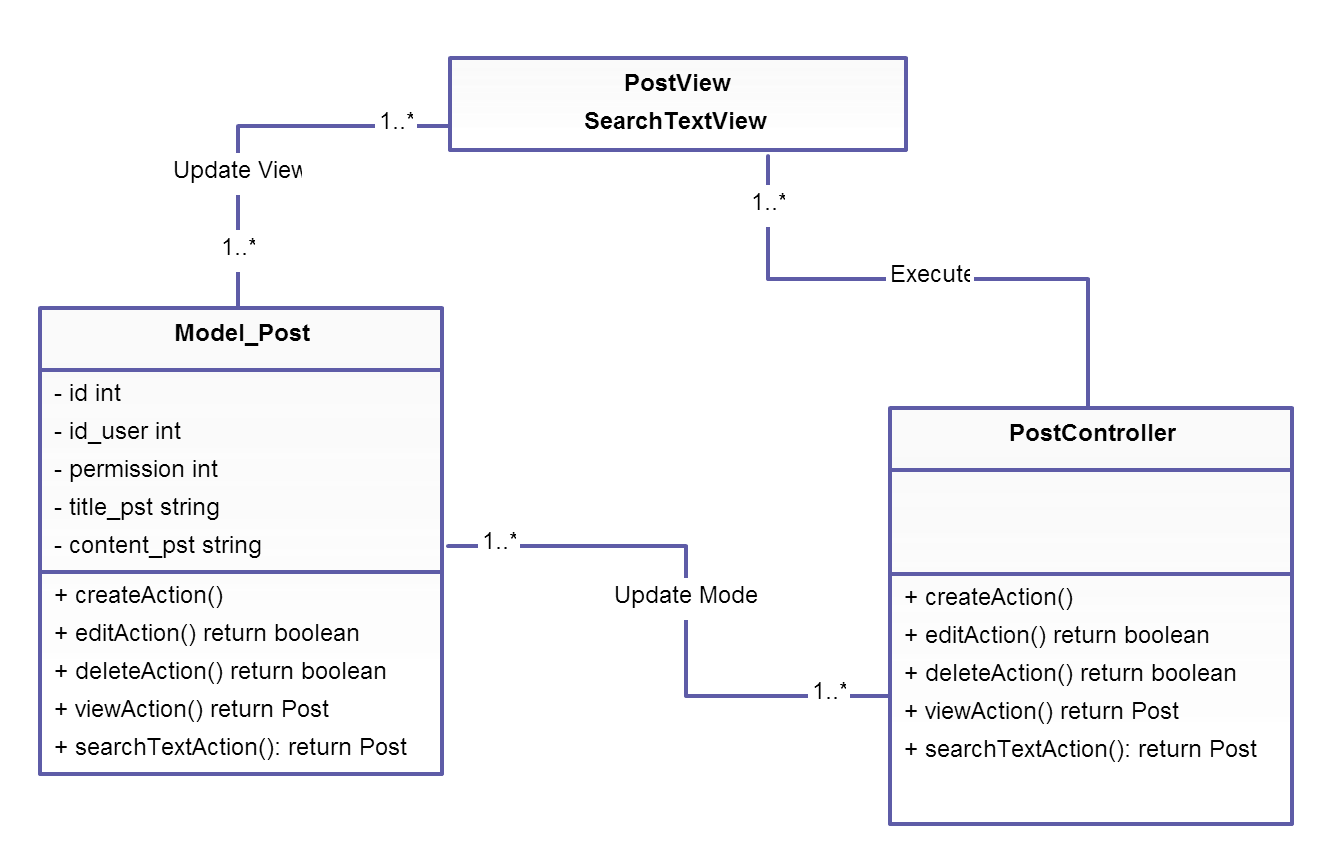


Figure 30: Post package

* Q&A package

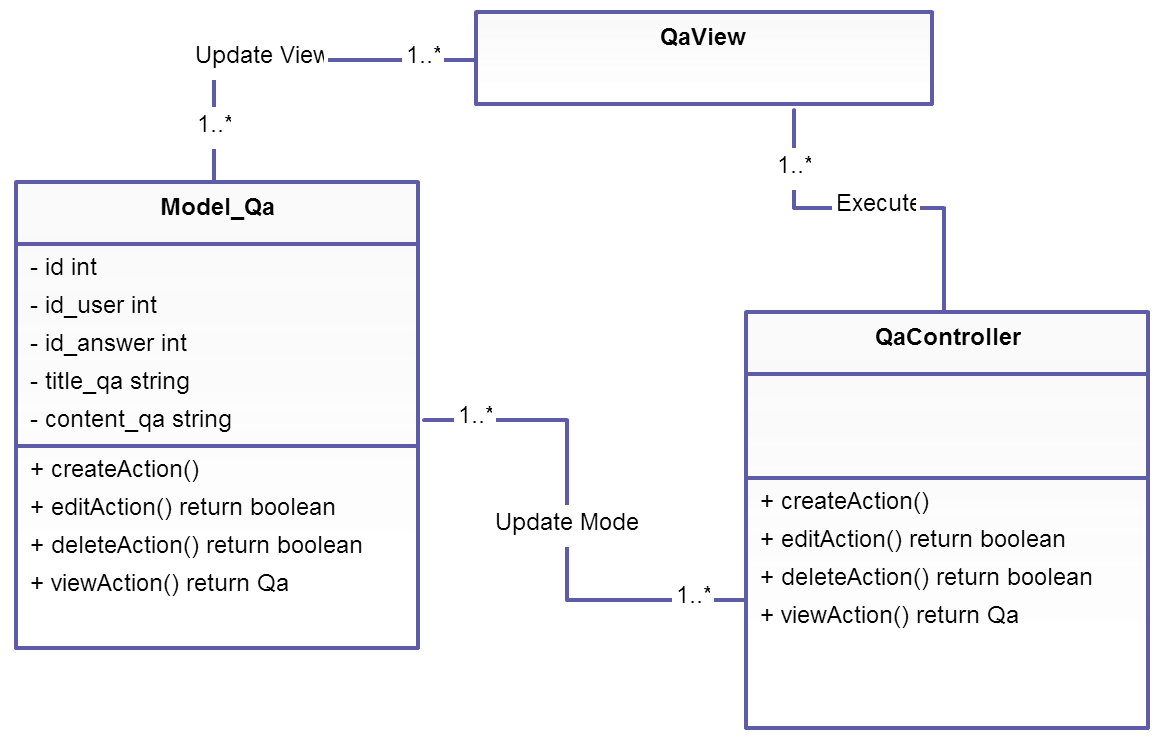


Figure 31: Q&A package

* Comment package

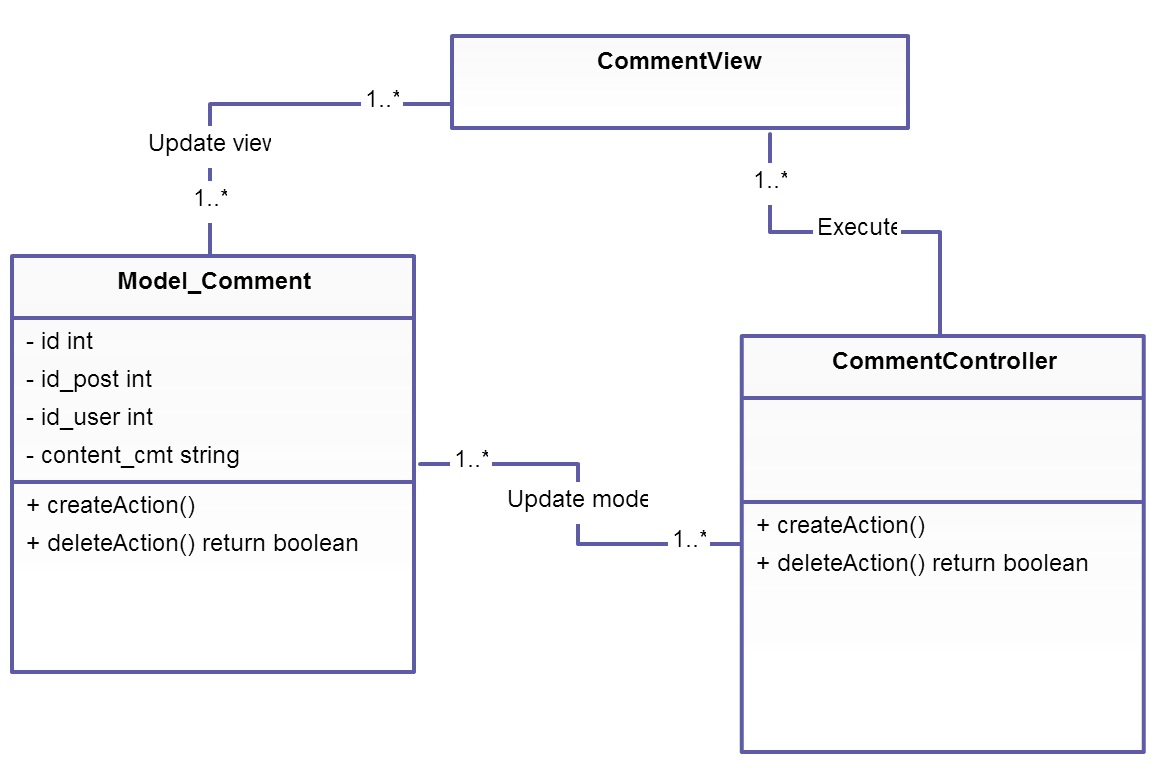


Figure 32: Comment package

* Admin package

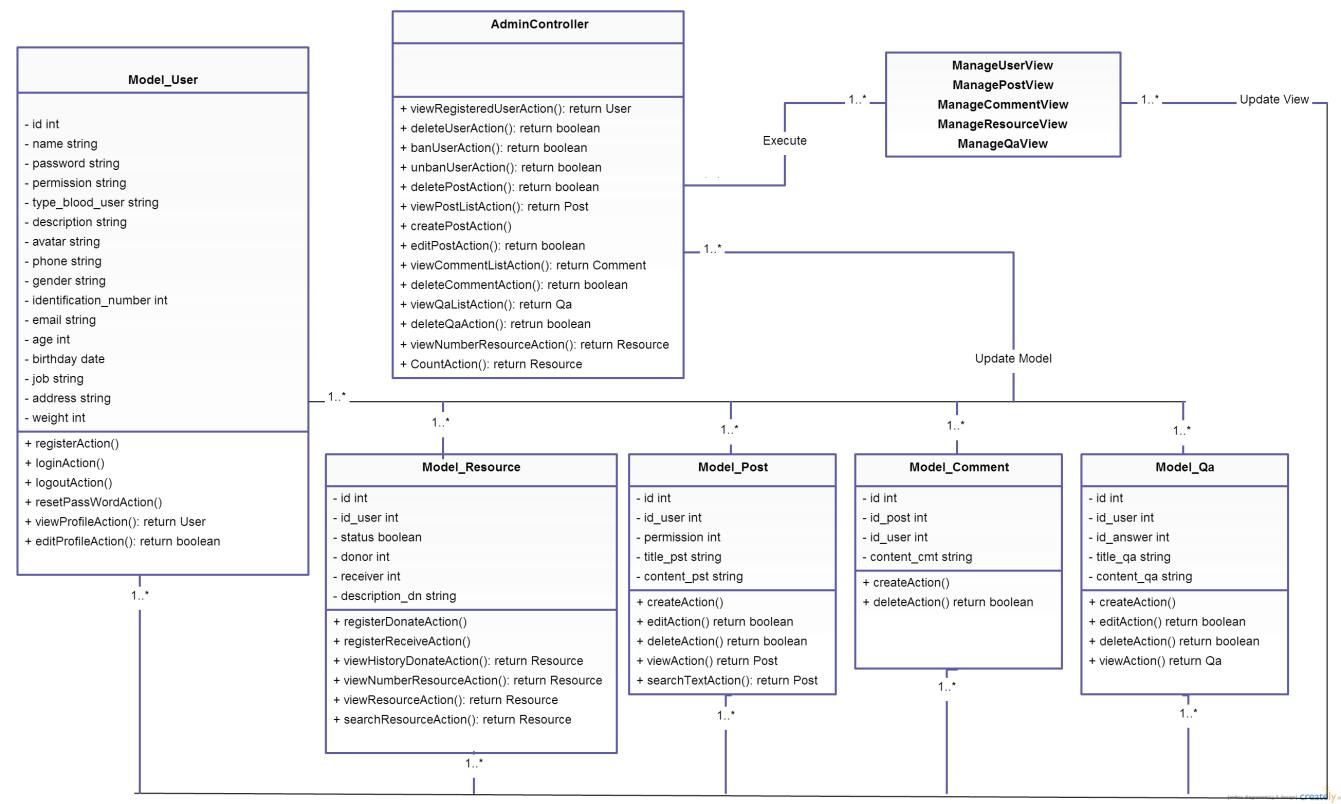


Figure 33: Admin package

## Layers

* Controller layer: Contain the interface between
* Associated models
* Associated views
* The input devices (e.g., keyboard, pointing device, time).
* Send commands to the model to update the model's state.
* Model layer is
  + the domain-specific software simulation
  + Or implementation of the application's central structure.
* View layer deal with everything graphical
* Requests data from their model
* Display the data

# Deployment view

Deployment view of website

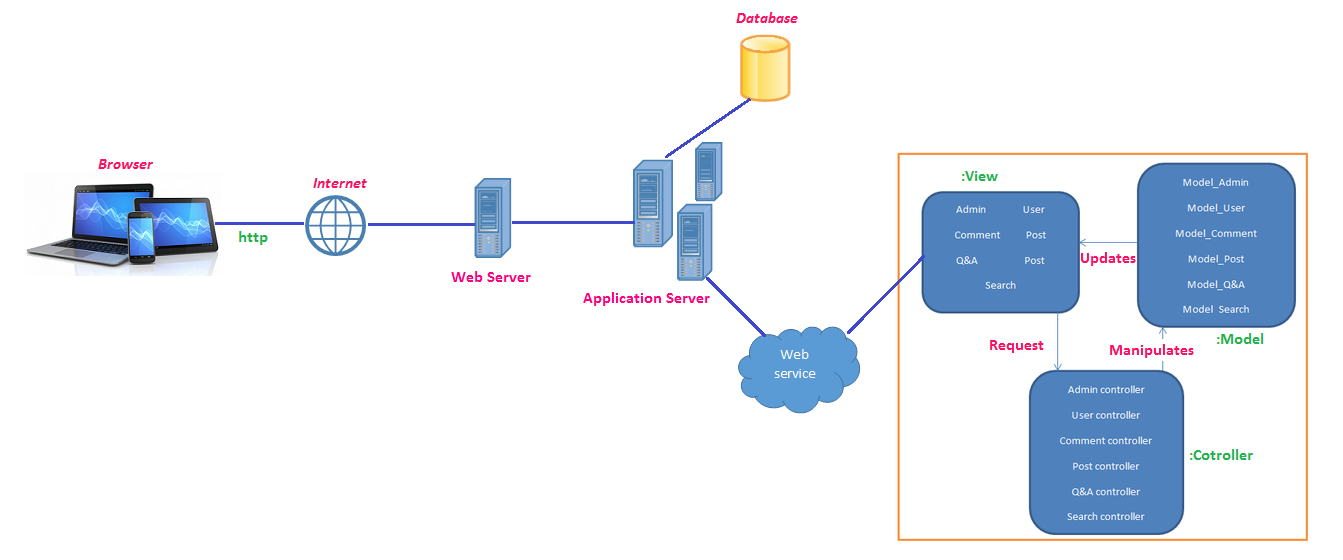


Figure 34: Deployment view

# Size and Performance

N/A

# Quality

* Scalability
* Description: Website’s reaction when user demands increase.
* Solution: Extend bandwidth for website.
* Reliability, Availability
* Description: Website’s reaction when user takes some action.
* Solution: Hire a good sever for website
* Security
* Description: Hide private information.
* Solution: Set access authorities to all users.

# Other Considerations

N/A