

USEFUL JAPANESE DICTIONARY FOR VIETNAMESE

Architecture Design

**Project Code: UJD\_VN**

**Document Code: UJD\_VN\_ Architecture Design\_v1.2\_EN**

**Ha Noi, 25/06/2014**

Record of change

\*A - Added M - Modified D - Deleted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Effective Date | Changed Items | A\* M, D | Change Description | New Version |
| 25/06/2014 |  | A | Add new | v1.0 |
| 07/07/2014 | Component diagram | M | Modified | v1.1 |
| 11/08/2014 | Sequence diagram | M | Modified | v1.2 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Table 1: Record of change

SIGNATURE PAGE

**ORIGINATOR:** Pham Tien Dat 26/06/2014

Developer

**REVIEWERS:**

**APPROVAL:**  Nguyen Van Sang

Supervisor

TABLE OF CONTENTS

[1 Introduction 5](#_Toc395831143)

[1.1 Purpose 5](#_Toc395831144)

[1.2 Scope 5](#_Toc395831145)

[1.3 Definitions, Acronyms and Abbreviations 5](#_Toc395831146)

[1.4 References 5](#_Toc395831147)

[1.5 Overview 6](#_Toc395831148)

[2 Architectural Representation 6](#_Toc395831149)

[3 Architectural Goals and Constraints 8](#_Toc395831150)

[3.1 Technical platform 8](#_Toc395831151)

[4 Use-Case View 9](#_Toc395831152)

[4.1 Use-Case Realizations 9](#_Toc395831153)

[5 Logical View 11](#_Toc395831154)

[5.1 Overview 11](#_Toc395831155)

[5.2 Architecturally Significant Design Packages 12](#_Toc395831156)

[6 Process View 13](#_Toc395831157)

[7 Implementation View 14](#_Toc395831158)

[7.1 Overview 14](#_Toc395831159)

[7.2 Layers 36](#_Toc395831160)

[8 Deployment view 37](#_Toc395831161)

[9 Size and Performance 37](#_Toc395831162)

[10 Quality 37](#_Toc395831163)

[11 Other Considerations 37](#_Toc395831164)

# 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 Useful Japanese Dictionary for Vietnamese website created by UJD\_VN capstone project team.

## Definitions, Acronyms and Abbreviations

|  |  |  |
| --- | --- | --- |
| Acronym | Definition | Note |
| UJD\_VN | Useful Japanese Dictionary for Vietnamese |  |
| MVC | Model view control |  |
| IDE | Integrated development environment |  |
| GUI | Graphic user interface |  |

Table 2: Definitions

## References

* UJD\_VN\_ Software Requirements Specification\_v1.0\_EN.docx
* UJD\_VN \_Data Design\_v1.0\_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>

## 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”.

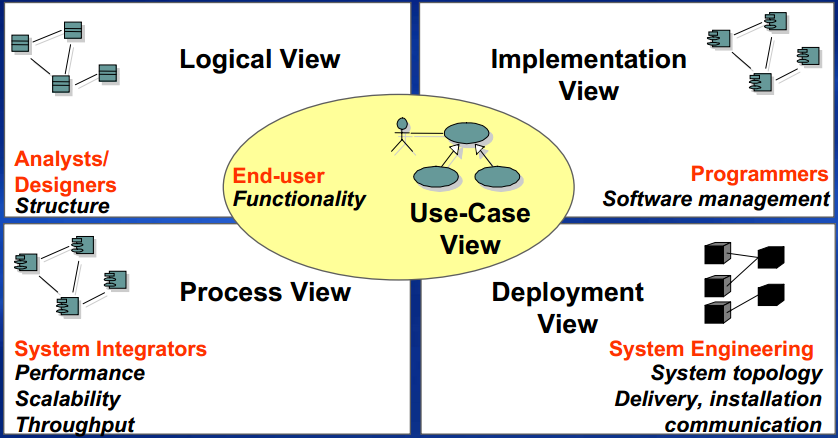


Figure 1: “4+1 view model”

|  |  |  |  |
| --- | --- | --- | --- |
|  | **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: Sublime Text 3
* Framework: Codeigniter framework.
* Web server: XAMPP 1.8.3.
* PHP 5.4.

# Use-Case View

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

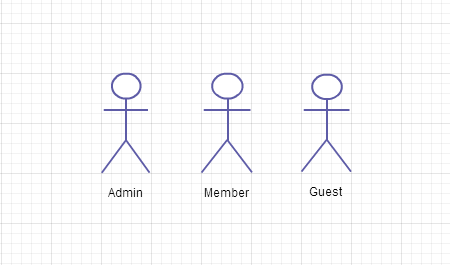


Figure 2: Actors of website

## Use-Case Realizations

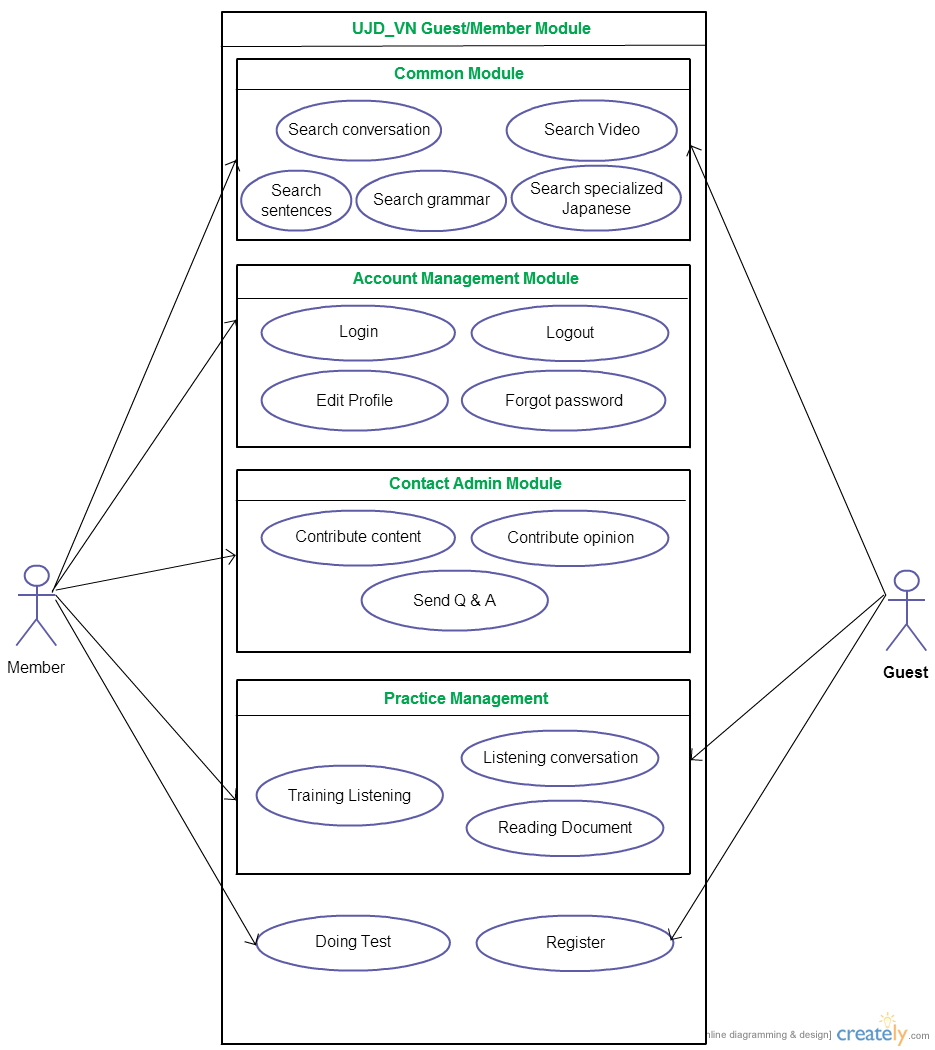


Figure 3: Member and Guest use-case

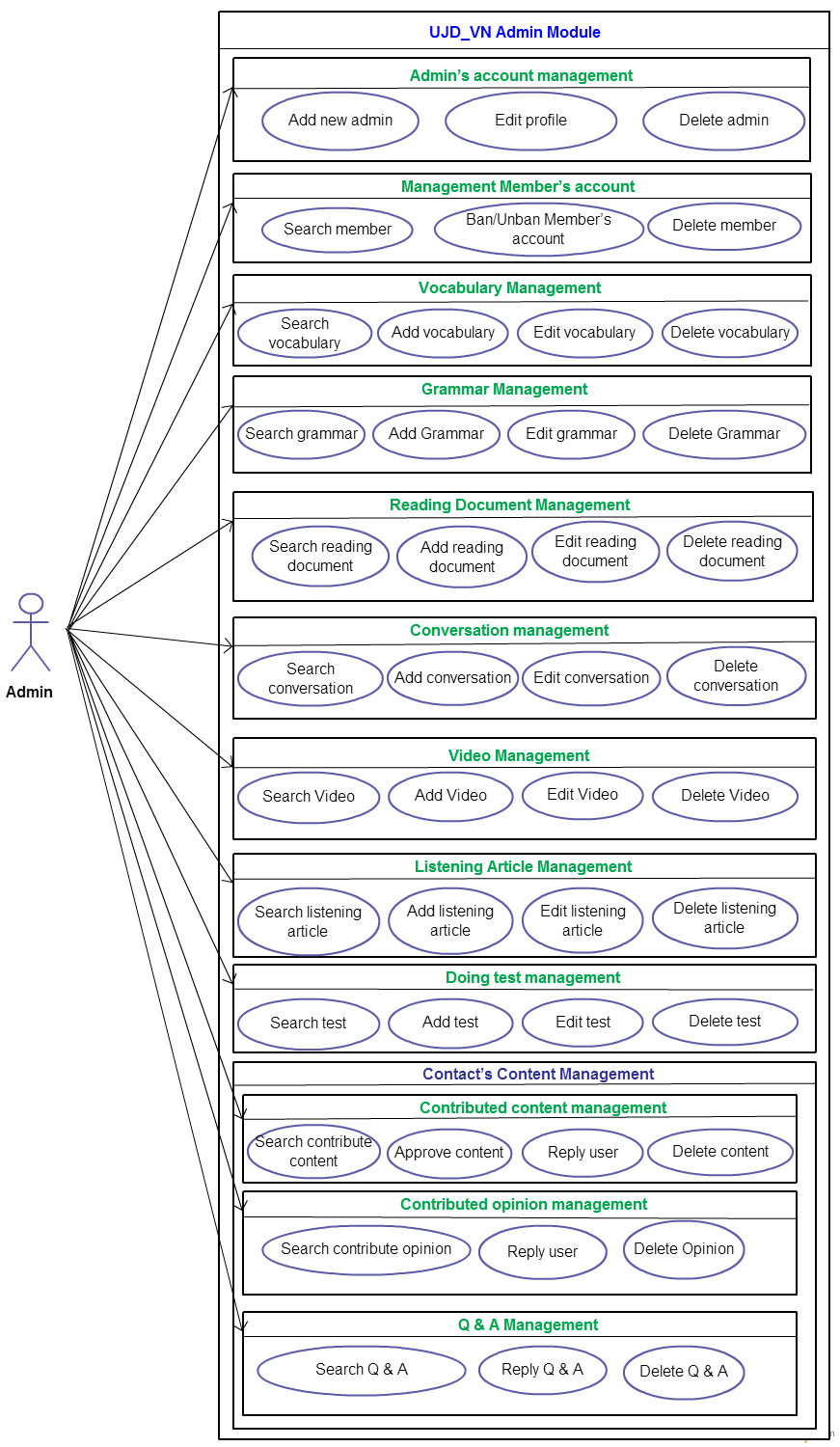


Figure 4: Admin use-case

# Logical View

This website based on Model view controller model.



Figure 5: 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

Public

GUI

User management

Q&A

Admin management

Business

Test management

Document management

Contributed management

Search

Data Access

Figure 7: System architect overview

Database Access

UJD\_VN (Mysql)

Database

# Process View

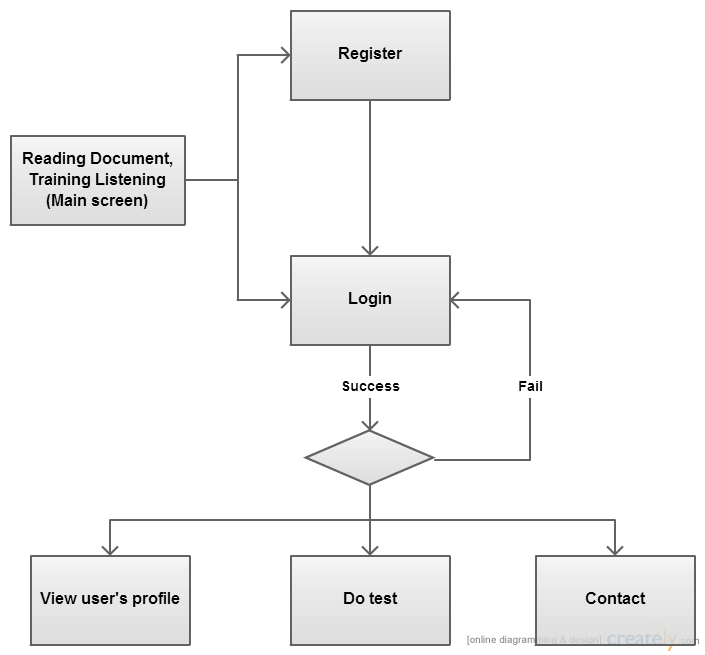


Figure 6: Process view

* Main screen: Process will start when guest connect to the website. In this process, user can reading document, training listening…Process end when guest click to login, register or close website.
* Login: When process starts website redirect to homepage. After user filling username, password and click login, system will search data in database. If success website will redirect to homepage. Process end when user logout or click do test/contact/view user’s profile function.
* View user’s profile : Pre-condition: User logged in.
* Process starts when user click to view profile, end when user click to other functions or close website.
* Do Test: Pre-condition: User logged in.
* Process starts when user click to test, end when user submit form test, click to other functions or close website.
* Contact: Pre-condition: User logged in.
* Process starts when user click to contact, end when user submit form contact, click to other functions or close website.

# Implementation View

## Overview

* Component diagram

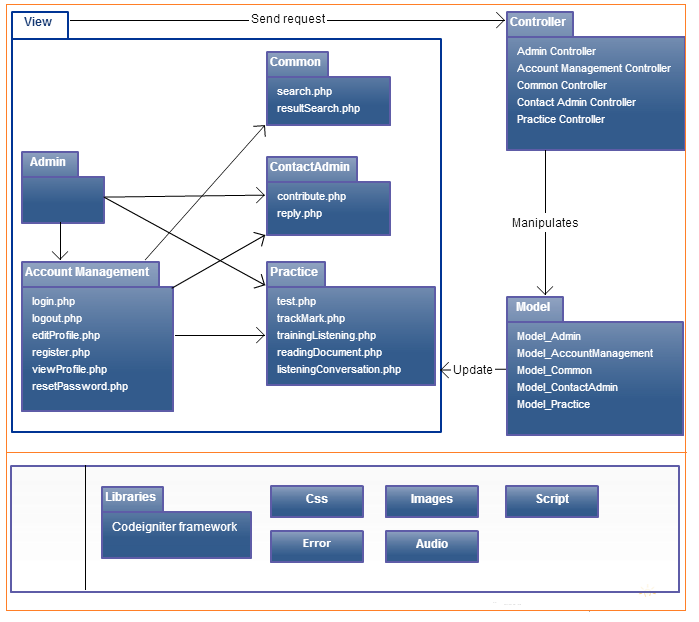


Figure 7: Component Diagram

* Sequence diagram

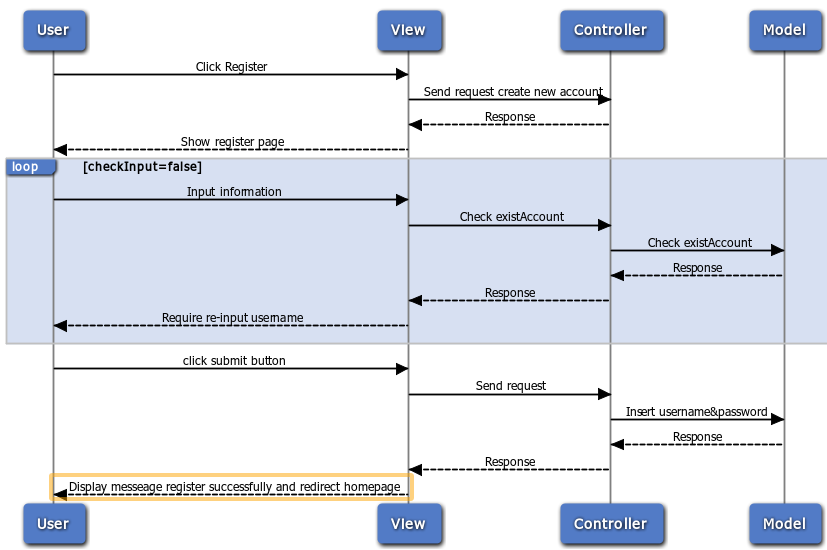


Figure 8: Register new account

* Reading Document

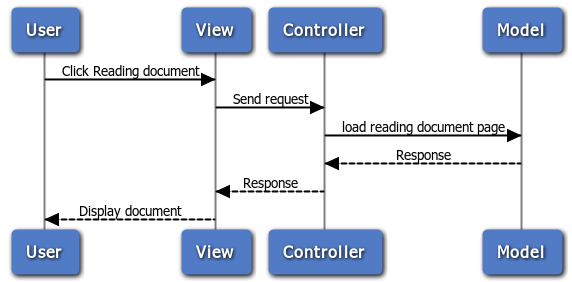


Figure 9: Reading Document

* Login

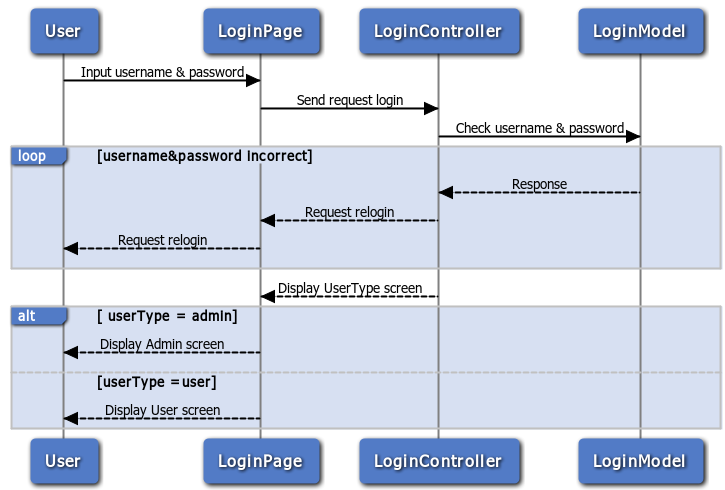


Figure 10: Login

* Logout

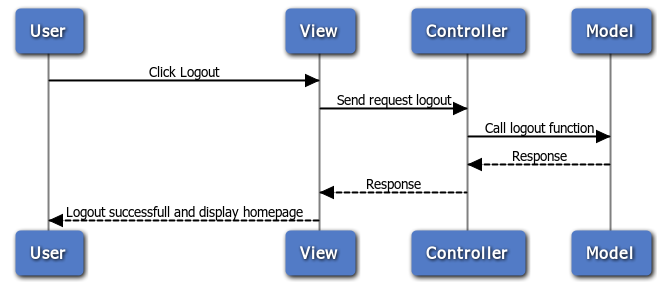


Figure 11: Logout

* Reset password

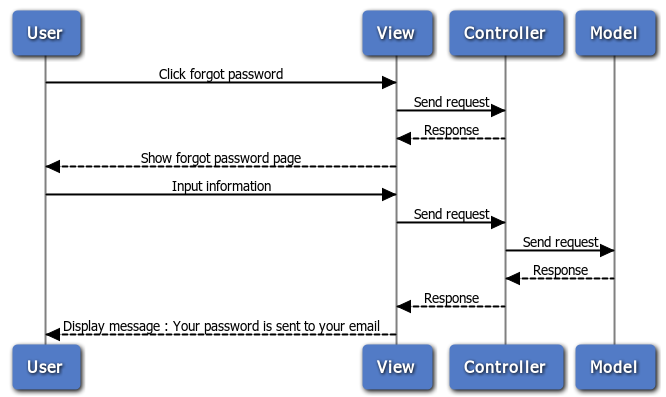


Figure 12: Reset password

* Edit profile

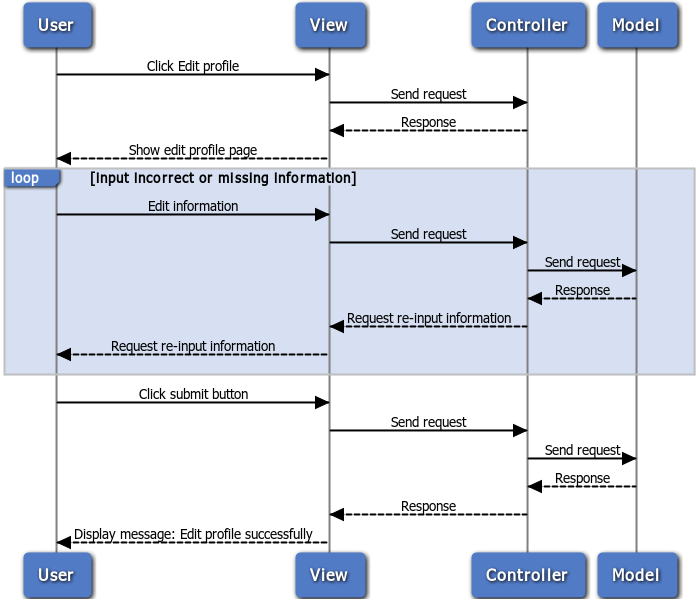


Figure 13: Edit profile

* Search

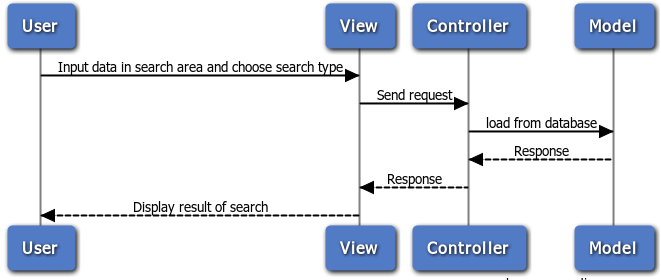


Figure 14: Search

* Do test

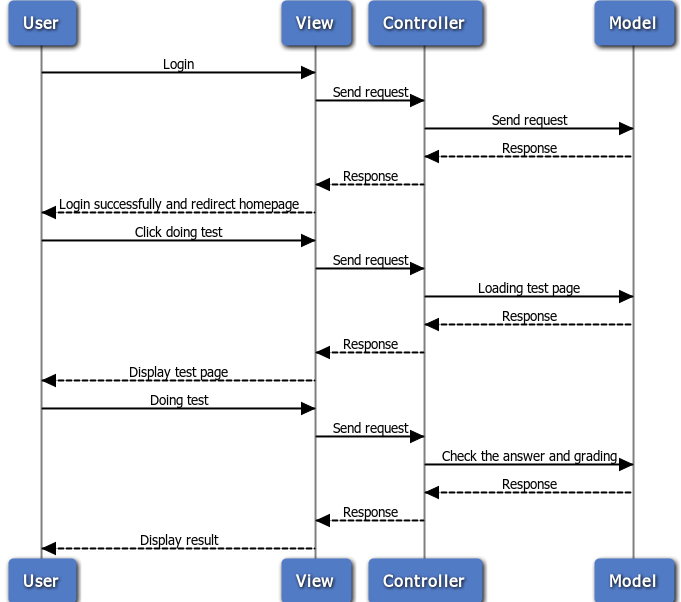


Figure 15: Do test

* Training Listening

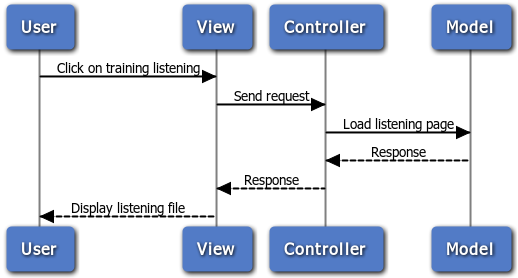


Figure 16: Training Listening

* Contact

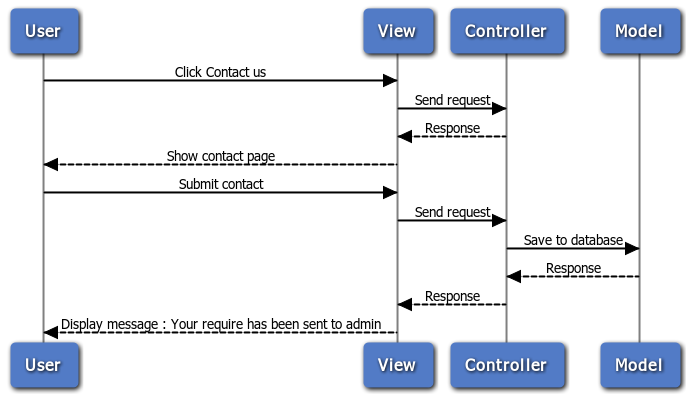


Figure 17: Contact

* Admin manage user

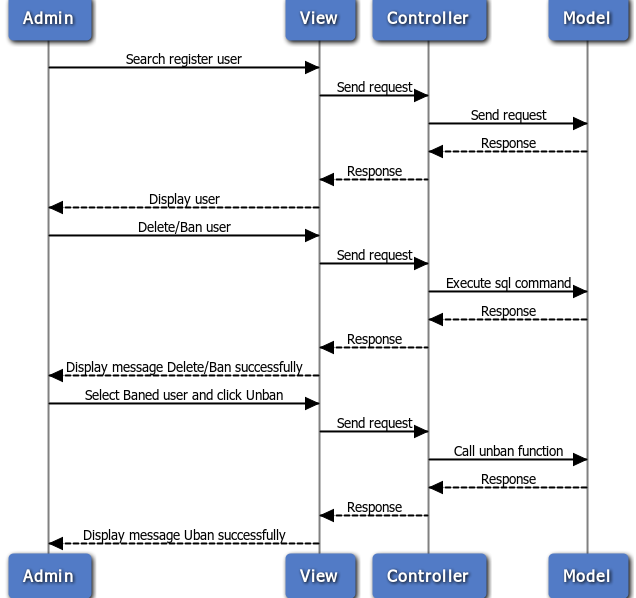


Figure 18: Admin manage user

* Admin manage vocabulary

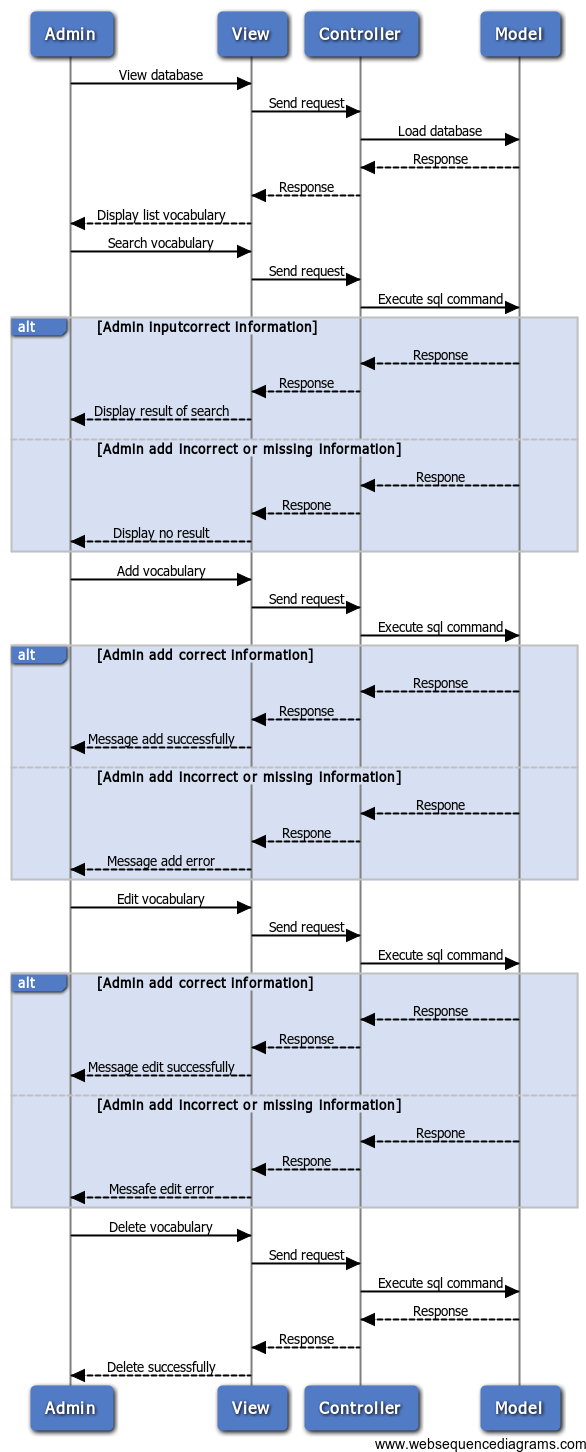


Figure 19: Admin manage vocabulary

* Admin manage grammar



Figure 20: Admin manage grammar

* Admin manage kanji

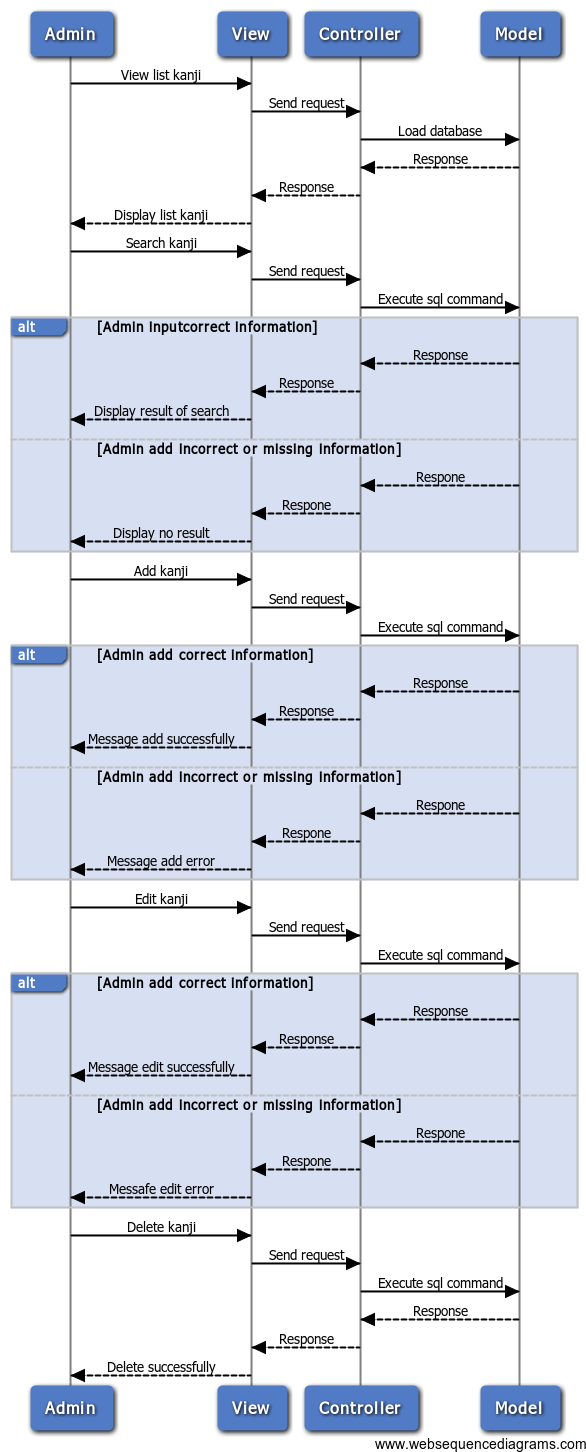


Figure 21: Admin manage kanji

* Admin manage video

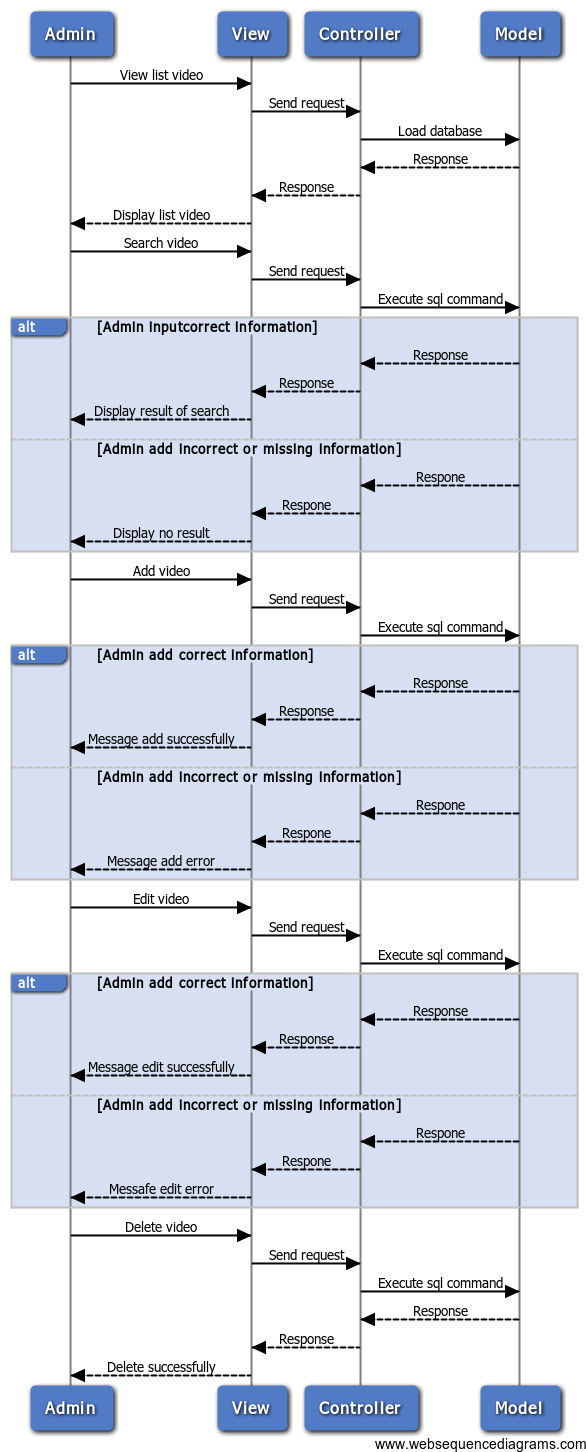


Figure 22: Admin manage video

* Admin manage conversation

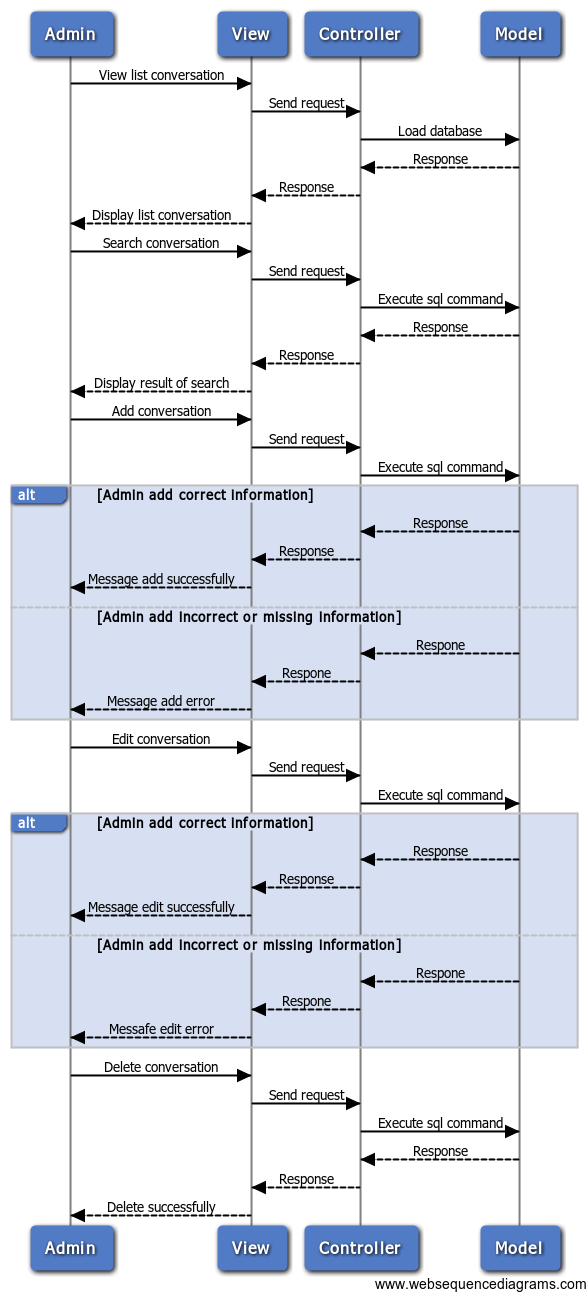


Figure 23: Admin manage conversation

* Admin manage reading document

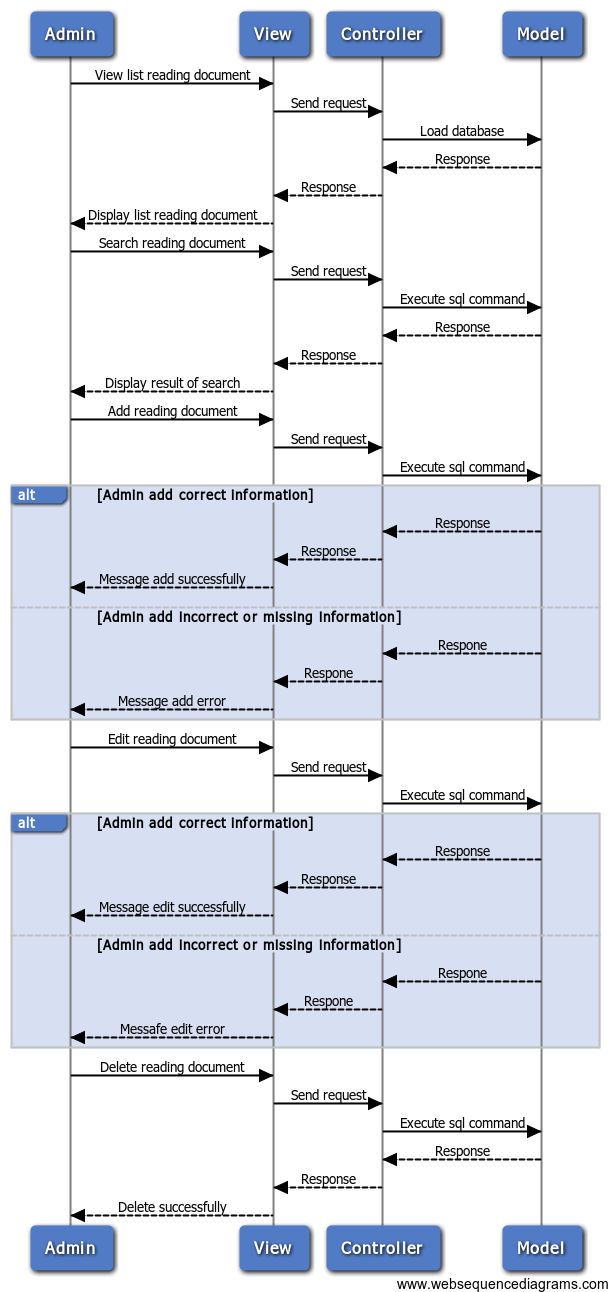


Figure 24: Admin manage vocabulary

* Admin manage training listening

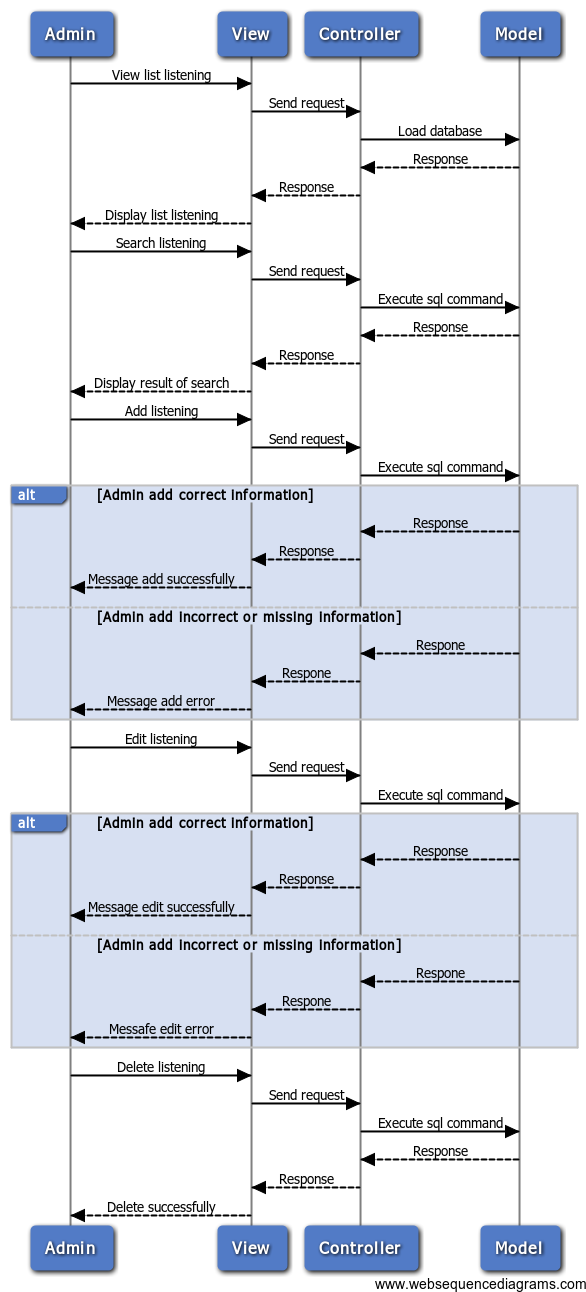


Figure 25: Admin manage vocabulary

* Admin manage test

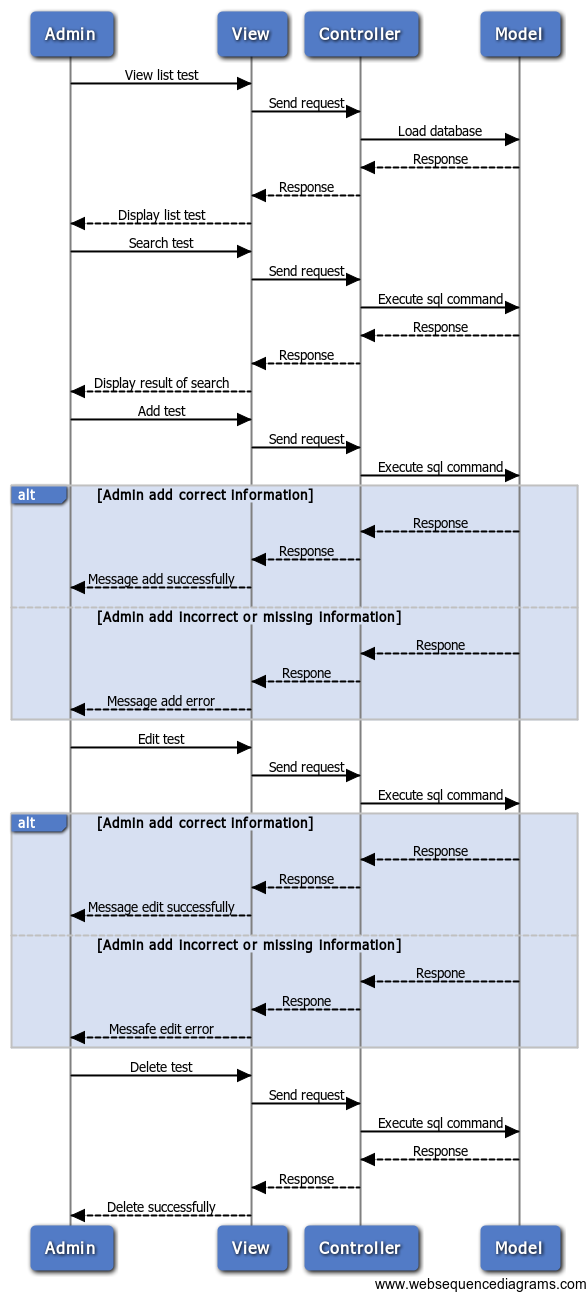


Figure 26: Admin manage vocabulary

* Admin manage contact

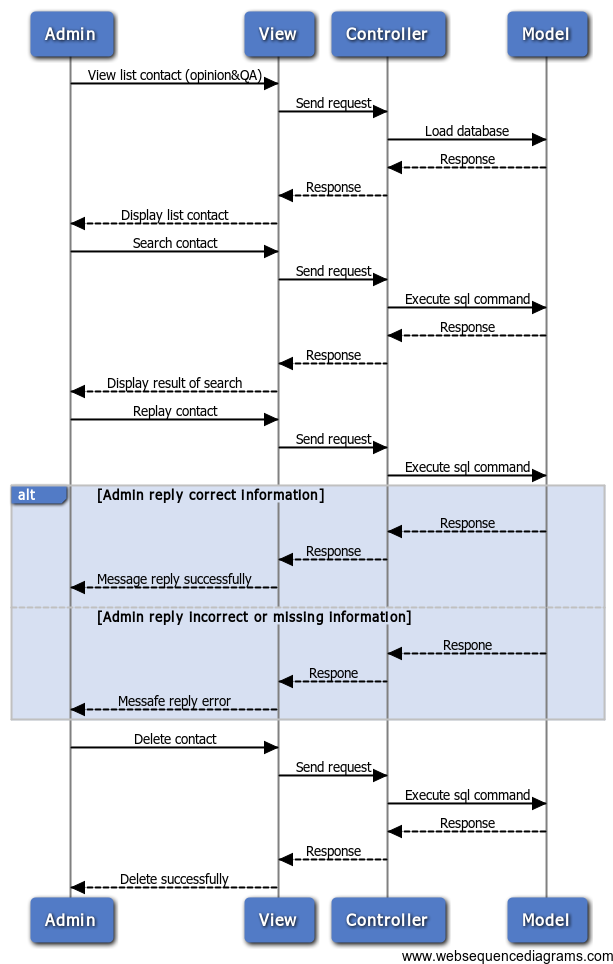


Figure 27: Admin manage vocabulary

* Class diagram
* User package

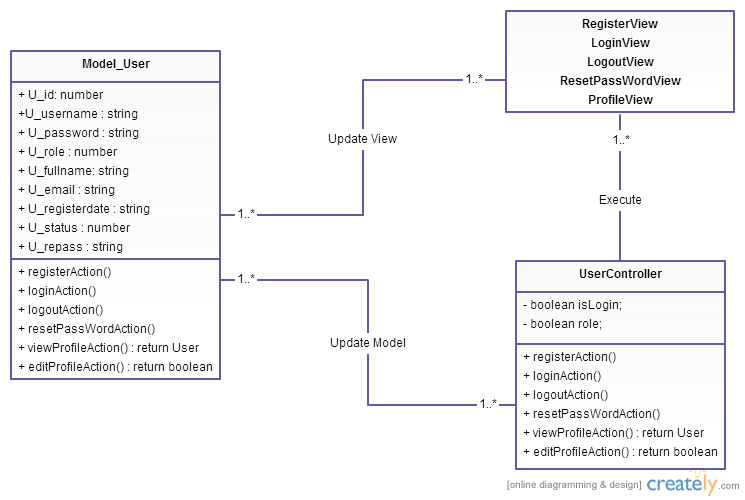


Figure 28: User package

* Admin package



Figure 29: Admin package

* Vocabulary package

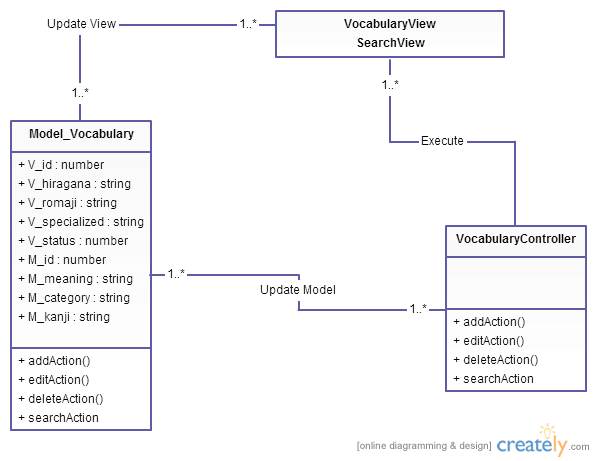


Figure 30: Vocabulary package

* Grammar package

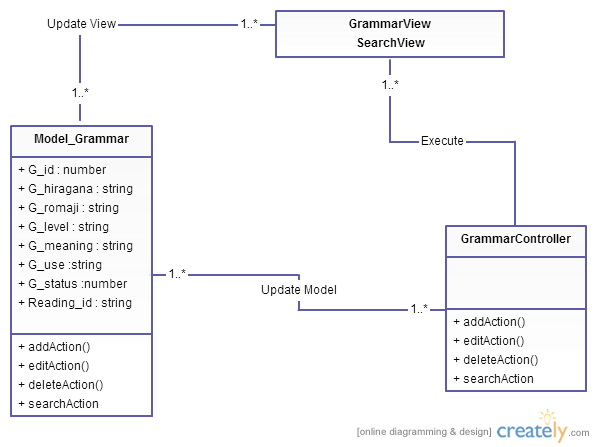


Figure 31: Grammar package

* Kanji package

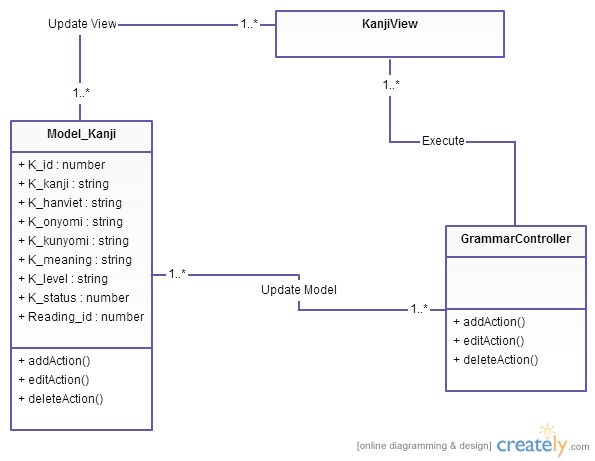


Figure 32: Kanji package

* TrainingListening package

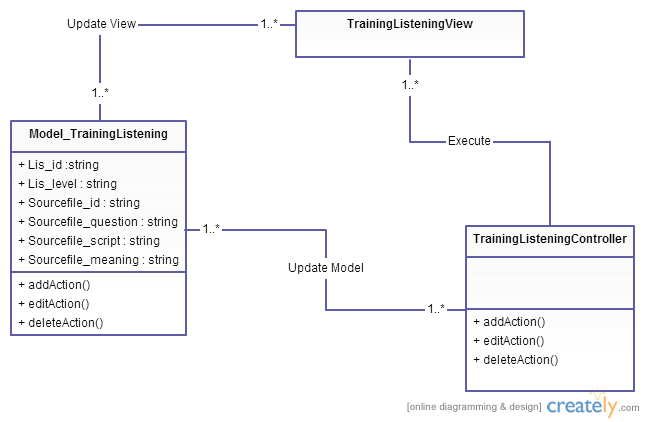


Figure 33: TrainingListening package

* Video package

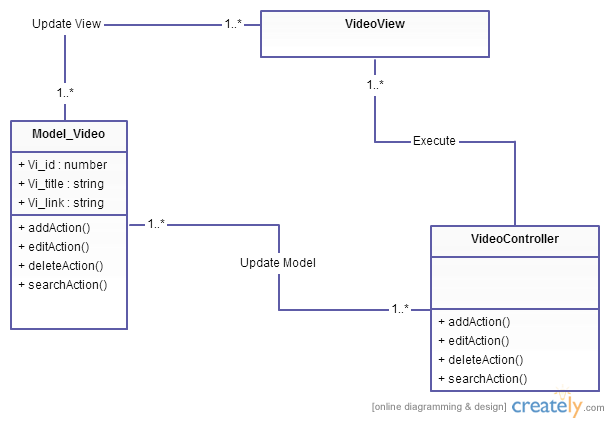


Figure 34: Video package

* Contact package

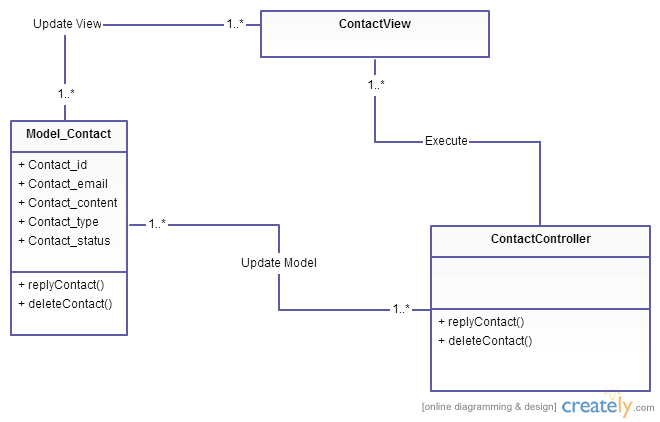


Figure 35: Contact package

* Sentence package

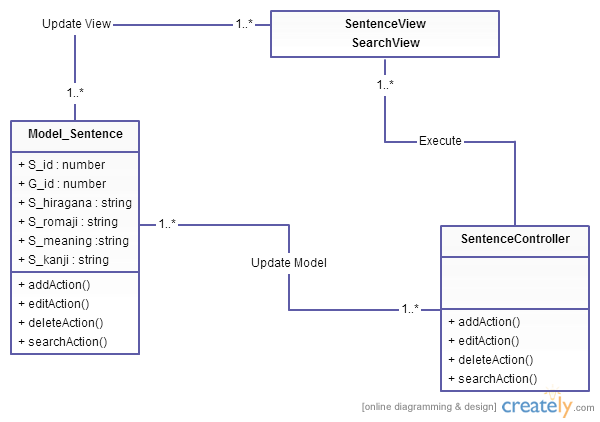


Figure 36: Sentence package

* Conversation package

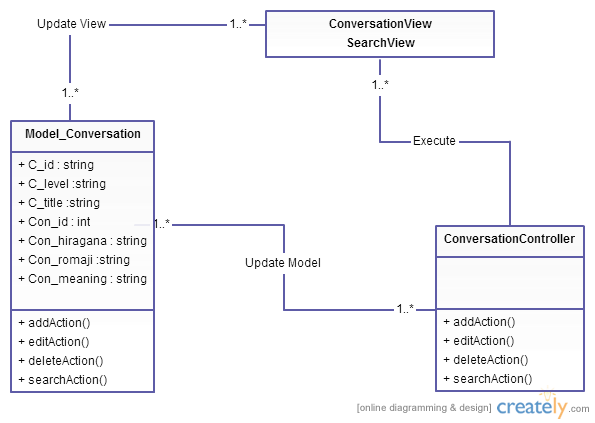


Figure 37: Conversation package

* Test package



Figure 38: Test 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

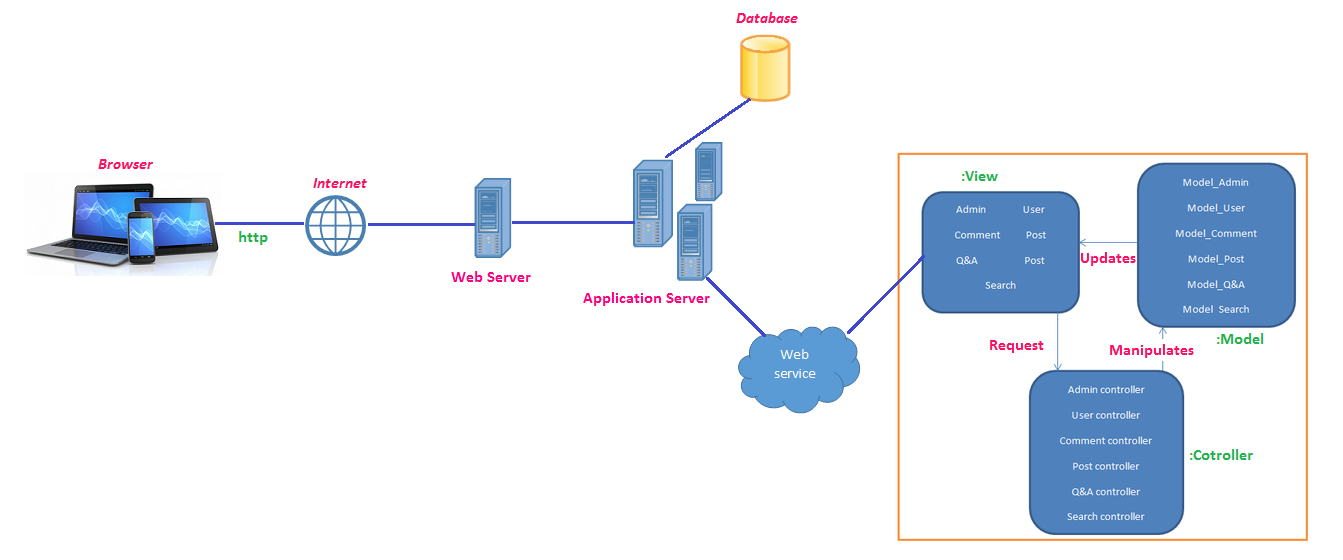


Figure 39: Deployment view

# 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