

**MEDICAL MANAGE**

**Software Design Document**

– Hanoi, February 2023 –

**Table of Contents**

[I. Overview](#_gjdgxs) **3**

[1. Code Packages/Namespaces](#_30j0zll) 3

[2. Coding Conventions](#_1fob9te) 3

[2.1. Naming Convention](#_mdyvgyhm4k5q) 3

[2.2. Comment Convention](#_mtlfj79yi0jk) 4

[II. Code Designs](#_3znysh7) **4**

[1. Common Function](#_2et92p0) 4

[2. <Feature/Function Name2>](#_2s8eyo1) 9

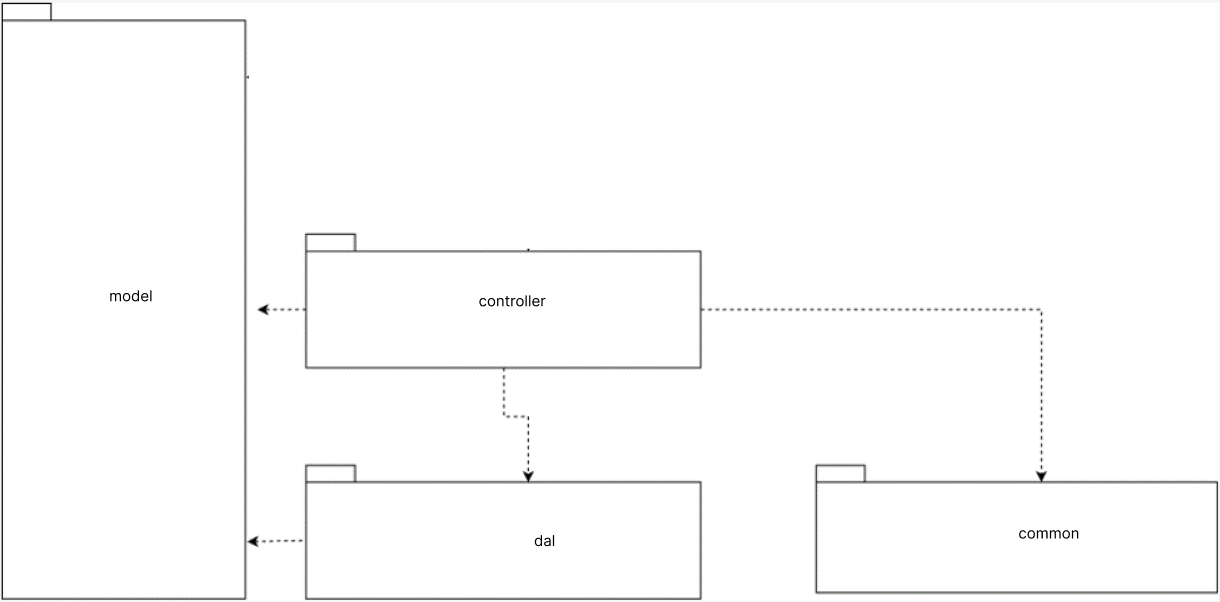
[III. Database Design](#_17dp8vu) **10**

[1. Database Schema](#_a4igkoo9da78) 10

[2. Table Description](#_pgvoyrdk31ar) 10

# I. Overview

## 1. Code Packages/Namespaces



***Package descriptions & package class naming conventions***

|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| *1* | *controller* | Act as an interface between Model and View components to process all the business logic and incoming requests |
| *2* | *common* | * Contain all classes that are used for exception handling, validation,.. |
| *3* | *dal* | Contain database connection and query |
| *4* | *model* | is used for components that correspond to all the data-related logic that the user works with |

## 2. Coding Conventions

### 2.1. Naming Convention

* **camelCase** is syntax where the first word of a phrase is written in lowercase and the characters of the following words are capitalized. Usually, this syntax will be used to write functions, variable names, and methods.
* **PascalCase** is the syntax that will be written in uppercase all the first characters in the word of the phrase. Usually, this syntax is used for class names, sometimes variable names.
* **snake\_case** syntax will have a common spelling that all letters are lowercase and separated by an underscore "\_". In some cases, in some languages, we have a variant of writing all letters in uppercase. Usually, we will see this syntax in constant naming and sometimes program names.
* Names should be descriptive of their function.
* Should not name abbreviations.
* Avoid generic names that don't make sense.

### 2.2. Wrapping Lines

When an expression will not fit on a single line, break it according to these general principles:

* Break after a comma.
* Break before an operator.
* Prefer higher-level breaks to lower-level breaks.
* Align the new line with the beginning of the expression at the same level on the previous line.