Utilizing Information System to Tackle Drug-Drug Interaction in Clinical Setting

1st Edition
November 2021

David Agustriawan, Ph.D. • apt. Audrey Amira Crystalia, M.Sc.
Dr.rer.nat. Arli Aditya Parikesit, S.Si., M.Si. • Andreas Whisnu, ST
Moch. Firmansyah, S.Si., MM • apt. Ike Veneqe, S.Si.
Margaretta Deidre Valeska, S.Bioinf.

Bioinformatics Departement • Pharmacy Departement
Indonesia International Institute for Life Sciences
Rumah Sakit Lira Medika

Utilizing Information System to Tackle Drug-Drug Interaction in Clinical Setting

David Agustriawan, Ph.D.
apt. Audrey Amira Crystalia, M.Sc.
Dr.rer.nat. Arli Aditya Parikesit, S.Si., M.Si.
Andreas Whisnu, ST
Moch. Firmansyah, S.Si., MM
apt. Ike Veneqe, S.Si.
Margaretta Deidre Valeska

i3L PRESS

Utilizing Information System to Tackle Drug-Drug Interaction in Clinical Setting

ISBN	:	
Page number	:	30

Authors	:	David Agustriawan, Ph.D.
		Indonesia International Institute for Life Sciences (i3L)
		apt. Audrey Amira Crystalia, M.Sc.
		Indonesia International Institute for Life Sciences (i3L)
		Dr.rer.nat. Arli Aditya Parikesit, S.Si., M.Si.
		Indonesia International Institute for Life Sciences (i3L)
		Andreas Whisnu, ST
		Indonesia International Institute for Life Sciences (i3L)
		Moch. Firmansyah, S.Si., MM
		Lira Medika Hospital
		apt. Ike Veneqe, S.Si.
		Lira Medika Hospital
		Margaretta Deidre Valeska, S.Bioinf.
		Indonesia International Institute for Life Sciences (i3L)

Editor	:	apt. Alya Ghina Aqila Arham, M.Sc.
Cover Design	:	Dimas Kurnia Pamungkas, B.Sc.
Published by	:	© i3L Press

Preface

Drug interaction, in particular drug-drug interaction (DDI), is one of the contributors of medication errors and adverse drug event incidence. DDI, although a preventable cause of medication error, were found to cause at least 30% of ADE in Pennsylvania, USA (Grissinger 2016). In Indonesia. DDI incident was also found to reached above 50% in one of the teaching hospital in Java (Rahmawati et al. 2006).

Preventing medication error and adverse drug event caused by drug interaction can be done by utilizing information technology and information system in healthcare practice. Many organizations have come out with a website or application for drug interaction checkers, such as IBM Micromedex®, Drugs.com, DrugBank, and Medscape. Unfortunately, some application or website is a paid service if healthcare practice wants to incorporate it to their electronic prescribing system with high subscription cost. In addition, those application and website not catered to drugs available in Indonesia, posing another difficulty especially for the hospital pharmacy department and hospital pharmacist.

This book was made to document the steps that we have taken to come up with innovative idea to tackle the DDI problems found in the hospital utilizing the experts in academia and the local hospital. This book contains step by step explanations on how to make innovation from drug-drug interaction problems in clinical practice in Indonesia. We hope that this book can inspire students to identify problems in healthcare and think of innovation to solve the problems. We also hoped through this book, students will be familiar with the strategies and steps that needs to be taken to implement the innovation idea that students have come up with. Lastly, we would like to express our appreciation to Lira Medika Hospital and Indonesia International Institute for life sciences for their continuous support in the implementation of the program. We also would like to thank the Ministry of Education, Culture, Research, and Technology of Indonesia for the funding of the program as a part of the Matching Fund scheme: 3717/E3/SPM-K.08/KL/2021.

Jakarta, November 2021

Authors

Table of Contents

MEDICATION ERROR, ADVERSE DRUG EVENT, AND DRUG INTERACTION

Medication error is one of the most important issues the clinical setting. Although data on medication error incident in primary care setting is difficult to obtain, approximately, 13% of adult patients and 31% of children patients were at risk of harm from medication error in the United States (Sears, Ross-White, & Godfrey, 2012). Moreover, in Indonesia, it was found that medication error was the most reported patient safety incident (Handayani, 2017).

The National Coordinating Council for Medication Error Reporting and Prevention define medication error as “any preventable event that may cause or lead to inappropriate medication use or patient harm while the medication is in the control of the health care professional, patient, or consumer. Such events may be related to professional practice, health care products, procedures, and systems, including prescribing, order communication, product labelling, packaging, and nomenclature, compounding, dispensing, distribution, administration, education, monitoring, and use” (National Coordinating Council for Medication Error Reporting and Prevention, 2015).

This definition suggest that medication error might be due, but not limited, to prescribing error, compounding and dispensing error, distribution of the medication, as well as administration and use of the medication. Prescribing error, in particular, is one of the major causes of medication error. A systematic review of the international literatures found that prevalence of prescribing error ranging from 2-94% (Assiri et al., 2018). It was found that in the United Kingdom 12% of primary care patients were affected with a prescribing or monitoring error and the incident was as high as 38% in elderly patients. In addition it was also found that 5% of all prescriptions had prescribing error (Avery et al., 2012). In Mexico, it was found that 58% of prescriptions contain error (Zavaleta-Bustos, Castro-Pastrana, Reyes-Hernández, López-Luna, & Bermúdez-Camps, 2008). These findings showed that medication errors are a global issues with high incidence both in developed as well as developing countries (World Health Organization, 2016).

Medication error as well as prescribing error could result in undesirable outcome including adverse drug event (ADE) and drug interactions (World Health Organization, 2016). Drug interactions in particular is one of the most common prescribing error. ADE is defined as “an injury resulting from medical intervention related to a drug” (Bates et al., 1995) which is more of a measure of harm to the patient as a result drug intervention. Meanwhile, drug interaction is a change in action or side effect of the drugs to concomitant administration with other drugs, food, beverages, as well as herbal supplement.

One of the most common causes of drug interactions in the clinical setting is the interaction between drugs, or also termed as drug-drug interaction (DDI). DDI is also one of the most common error found in prescribing and dispensing (Nurfitria, Adillah Effendi, & Iskandar, 2019). DDI can results in increased or decreased pharmacological effect as well as an adverse reaction from a drug. DDIs are also deemed to be one of the most common cause of ADE (Palleria et al., 2013). This phenomenon was estimated to comprise of at least 30% of all adverse drug events (ADE) in Pennsylvania, USA (Grissinger 2016). It was also reported in Canada that prevalence of DDI reached as high as 18.7% in patients who received at least 2 infusion drugs (Grissinger 2016). In Indonesia, drug interaction prevalence in one of the teaching hospitals in Yogyakarta was found to be as high as 59% and 69% in inpatients and outpatients, respectively. Among the findings, DDI comprised of 38.4% and 36.7% from total drug interaction found in inpatients and outpatients, respectively (Rahmawati *et al.* 2006).

DDI has been linked to ADE that cause prolonged hospitalization or new hospital admission (Dechanont *et al.* 2014). In group of patients who received multiple drugs, the risk of ADE caused by DDI is substantially increased (Cascorbi, 2012). Consequences of DDI also include additional cost of treatment to manage the ADE or decreased of drug pharmacological effect. Unfortunately, DDI data in Indonesia is still lacking most likely due to the lack of national reporting system on DDI as well as lack of resources and technology to identify DDI in most healthcare system in Indonesia.

Considering the possible harm caused by DDI, it is essential to prevent DDI that is a part of medication error to ensure patient safety. Studies found that using automated information system managed to reduce incident of medication error (World Health Organization, 2016). Computerized provider order entry (CPOE) with decision support might be effective to reduce the number of medication error, especially that is caused by DDI. To tackle this problem, in addition to educate healthcare providers and patients as well as implement medication review and reconciliation, The WHO recommend healthcare practice to implement and strengthened electronic prescribing and alert system (World Health Organization, 2016) which can be in a form of CPOE with DDI alert system focusing on clinically-relevant warnings.

INNOVATION PIPELINE: DRUG-DRUG INTERACTION DATABASE

Knowing that DDI is one of a major problem in medication error and patient safety issue, it is important to try to find a solution for such problem in Indonesia. To propose the solution, steps can be taken as displayed in figure 1 below.

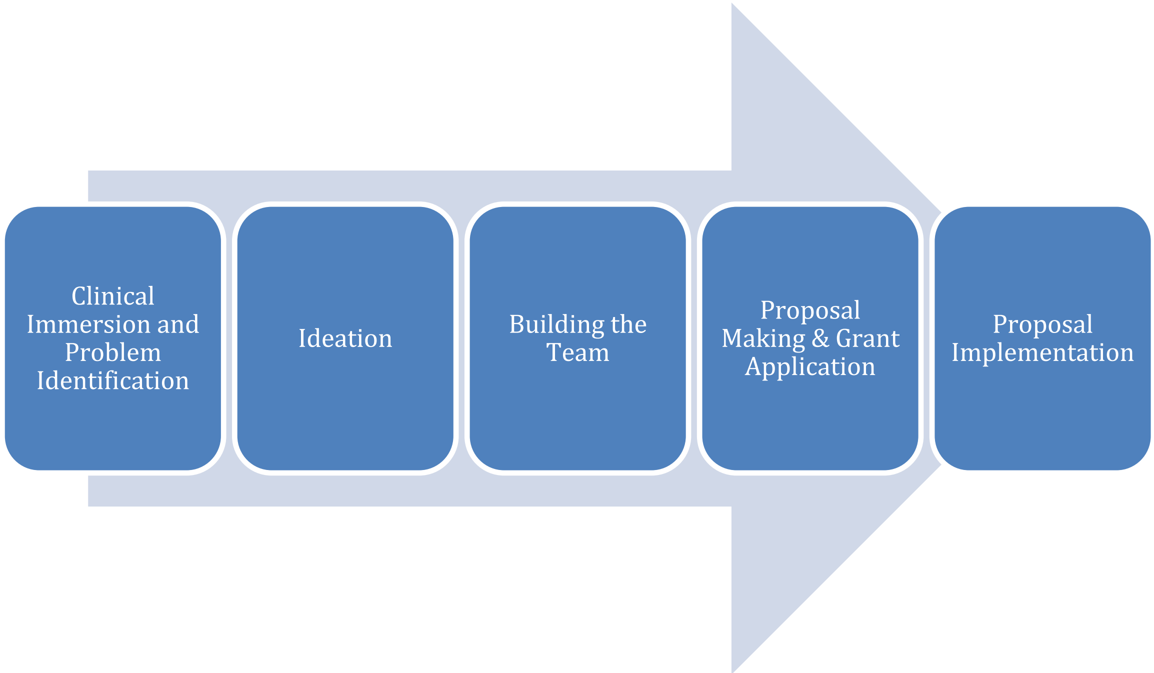


Figure 1. DDI Database Innovation Pipeline

Step 1: Clinical Immersion and Problem Identification

Following the pipeline, to propose solution problem identification is the necessary first step. From the literature, we identified that DDI is a major problem in ensuring patient safety. Furthermore, it was pretty clear that utilizing automatic information system would be ideal to tackle the problem. We conducted a research of the commercially available drug interactions databases and programs, and some free and paid services were found as follow:

- IBM® Micromedex®
- Lexicomp©
- Medscape (https://www.medscape.org)

- Drugs.com (https://www.drugs.com)
- DrugBank (https://go.drugbank.com)

From the databases listed above, one problem is that the databases are not freely available (IBM® Micromedex® and Lexicomp©) and requires paid subscription. Another problem for the web-based freely available databases such as Medscape, Drugs.com, and DrugBank, are the limitation of such databases to be incorporated to local CPOE system in the hospital to allow for alert system. Free databases are also often used in the hospital pharmacy department as a tool for drug interaction checker and one major problem with using the system separately with the CPOE is that the databases are mostly based on FDA approved list of drugs and the searches are using the generic name of the drugs. This can pose a problem in efficiency of drug preparing and dispensing process by the pharmacy department in the hospital as most physicians prescribed drugs using its brand name. In addition, difficulty in making data comparison with locally available drug products in Indonesia can cause another error while checking for DDIs.

Having this information on hand, we conducted a clinical immersion with our partner Lira Medika Hospital to study more about the hospital needs in regard to DDIs identification to prevent medication error. During the clinical immersion, a lot of meetings were conducted with the information technology (IT) and pharmacy team from Lira

Medika Hospital. It was found that although the utilization of paid subscription from IBM® Micromedex® or Lexicomp© by incorporating the database to local CPOE system is one of the options to increase the efficiency in checking DDIs from physician's order and prevent medication error, the price was very high. In addition, at the time of clinical immersion, the hospital used Medscape to check possible DDIs from physicians' medication order and this significantly affect the order response time of the pharmacy department. The response time was longer due to the additional steps of inputting all the prescribed items from each prescription to Medscape to check for possible interaction. Not to mention, when possible DDI came up, the pharmacist would need to contact the prescriber and alert the physician of the possible DDI as well as confirm the prescription. This inefficiency further posed a problem when the prescription load was high which lead to errors in the DDI checking step as well as longer time needed to dispensed medicine which can also lead to medication error due to late administration of drug.

Following the clinical immersion process, the team identified that the lack of freely available DDIs databases which catered to Indonesia's approved list of drugs to be incorporated to local CPOE system is the problem that we are facing in the hospital. From here we move one to the ideation step on how we can tackle this problem.

Step 2: Ideation

In the ideation step, it was pretty apparent that we will need to create a database of our own that can be linked to the hospital CPOE system. For this database and system to be fully integrated, we found that creating a web-based application can work as the intermediary. This website will be used to manage both the system and database, as it will be able to manage the users who will be using the system and it will also manage the drug and DDI datasets contained inside the database.

The database created will need to cater specifically to Indonesian drug products, so our initial dataset will come from drug product inventory provided by Lira Medika Hospital. DDI information to complement the initial dataset will come from DrugBank as they have a freely available DDI dataset that can be downloaded by anyone. To finalize our database, both Lira Medika and DrugBank datasets will be cross-referenced with each other, and the final dataset will form the main content of our drug database. It is important to note that this database that will be created will differ from other databases available that have been mentioned previously, because it will provide information for both drug product name (brand name) and drug contents (generic name), instead of being limited by one type of information. Our database will also have DDI information of drug contents that can be used by healthcare professionals during prescription ordering step and also during the preparing and dispensing process in the hospital pharmacy.

We also decide to add a DDI checker to the integrated website application to provide users with an immediate tool during prescription ordering as well as drug preparing and dispensing process. This DDI checker will work similarly like a simple search tool (e.g. Google), where the user simply inputs the drugs they wish to look into in the search boxes, whether using the generic or brand name of the drug, then the system will look into the DDI of the queried drugs inside the database and immediately show the DDI as the output to the user. This DDI checker will be able to increase efficiency during the drug preparing and dispensing process since healthcare professionals will not have to use other tools or databases with our integrated drug system & database.

Step 3: Building the Team

To be able to execute the idea, it is essential to have a team with various expertise. Because the problem itself is about DDIs and the hospital pharmacy operations, we included expert from pharmacy specifically in clinical pharmacy and pharmacology as well as expert in hospital pharmacy from our hospital partner. These two members would allow the team to validate data in the database as well as implement the new system in the hospital.

The backbone of this innovation is information system and information technology. To enable us to create the database as such that the database allows effective and efficient usage whether it is by the public or incorporated to the hospital CPOE with alert system in place, the team needs experts in bioinformatics, programming, and information system. Those experts should also be joined by IT expert from the hospital that will help with incorporation of the databases to the local system.

Step 4: Proposal Making & Grant Application

As with every innovation, this project also needs funding to run. When the team have been formed, proposal making and submission to grant application is the essential next step. This step started with identifying potential grants to apply this project to.

Several grants that could be a potential fund sources for this type of project are listed in Table 1 below.

Table 1. Funding Sources for DDI Innovation Project

No.	Funding Body	Funding Scheme
1	Ministry of Education, Culture, Research, and Technology (*Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi*)	Matching Fund (https://kedaireka.id)
2	Australia Awards in Indonesia	Alumni Grant Scheme
3	Ministry of Education, Culture, Research, and Technology (*Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi*)	Riset Keilmuan (https://beasiswadosen.kemdikbud.go.id/risetkeilmuan)
4	Internal funding from i3L	LPPM Scheme for Internal Funding (CSR and/or Innovation Scheme)

Identification of which funding scheme is suitable for the project will also dictates on how the proposal is made. In making the proposal, every funding body and funding scheme might have different format and different requirements that should be met. However, in general, each proposal should cover:

1. Background and rationale of the project

This section should explain the background and rationale of the project which include problem identification. This section should be able to explain the importance of the project or program proposed.

2. Objective

   This section should include the objective and aim of the proposed program.

3. Impact and Possible Outputs

   This section should explain the quantitative and qualitative impact of the program as well as all the possible outputs of the program (i.e. patent, publication, copyright, etc.).

4. Program Design and Roadmap

   This section should include the design of the program that is proposed in the proposal and the roadmap for future development from the program. If there is any prior program directly related to the proposed program, it also should be included in the roadmap.

5. Methods of Program Implementation

   Step by step explanation on how the proposed program will be implemented.

6. Budget

   Budget needed to run the program, which include consumables, equipment, and cost for human resources. The budget should also factor in any cost related to program output. Budget should strictly adhere to the funding scheme requirement and specification.

7. Timeline

   Timeline needed for program implementation which usually are divided into terms or months. The timeline should be able to specifically show working period for each step in the program implementation.

Step 5: Proposal Implementation

After the funding is granted, now it is time for proposal implementation. This step is where the program design and methods included in the proposal are executed. In this step, it is important to keep in mind the timeline and outputs being promised in the proposal as the outputs should be fulfilled by the end of the funding period.

Following the 5 steps explained above, the authors, as a team, managed to secure funding from Matching Fund to implement our innovation program on creating DDI database which catered to Indonesia’s approved list of drugs and can be incorporated to local hospital CPOE system. The making of the system will be explained in the next section of this book.

Drug Interaction Database and Its Application in Clinical Setting

This section will discuss the process in creating the drug interaction system and the explanation on all the pages created in the system.

Creating a Database for The System

The database was created using MariaDB 15 software where the documentation can be found in this link https://mariadb.com/kb/en/documentation/. Below are the list of the tables with their entities that created for the drug-drug interaction system.

1. **Table Users**

   This table is used to store the user information, both for i3L'user and Lira Medika Hospital's user.

Name	Type	Length	Descriptions
idusers	bigInt	10	
idhospitals	int	10	
name	varchar	75	
email	varchar	255	
password	varchar	255	
avatar	varchar	100	
login_count	int	10	
created_by	int	10	
updated_by	int	10	
created_at	timestamp	0	
updated_at	timestamp	0	
deleted_by	int	10	
deleted_at	timestamp	0	

2. **Table Roles**

   This table is used to store *role permission* information which will be joined to the “*role_users*”.

Name	Type	Length	Descriptions
idroles	bigInt	20	
name	varchar	255	
slug	varchar	255	
permissions	longtext	0	
created_by	int	10	
updated_by	int	10	
created_at	timestamp	0	
updated_at	timestamp	0	

3. **Table Role Users**

   This table is used to store the relation between *users* and *roles*.

Name	Type	Length	Descriptions
idroleusers	bigInt	20	
idusers	bigInt	20	
idroles	bigInt	20	
created_by	int	10	
updated_by	int	10	
created_at	timestamp	0	
updated_at	timestamp	0	

4. **Table Hospitals**

   This table is used to store Lira Medika hospital’s information.

Name	Type	Length	Descriptions
idhospitals	bigInt	20	

name	varchar	50	
address	varchar	150	
type	varchar	100	- Publicly owned hospitals - Nonprofit hospitals - For-profit hospitals
logo	varchar	250	
created_by	int	10	
updated_by	int	10	
created_at	timestamp	0	
updated_at	timestamp	0	
deleted_by	int	10	
deleted_at	timestamp	0	

5. **Table Products**

   This table will be used to store product dataset owned by i3L which will be used by RS Lira Medika.

Name	Type	Length	Descriptions
idproducts	bigInt	20	
products_code	varchar	255	
name	varchar	250	
created_by	int	10	
updated_by	int	10	
created_at	timestamp	0	
updated_at	timestamp	0	
deleted_by	int	10	
deleted_at	timestamp	0	

6. **Table Contents Products**

   This table will store the composition of the product where the ID drug is referred to the other table.

Nama	Type	Length	Descriptions
id	bigInt	20	
idproducts	bigInt	20	
iddrugs	varchar	255	
nama	varchar	255	
description	text	0	
created_by	int	10	
updated_by	int	10	
created_at	timestamp	0	
updated_at	timestamp	0	
deleted_by	int	10	
deleted_at	timestamp	0	

7. **Table Request Products**

This table will be used to store the product dataset which is requested by Lira Medika hospital, when the product is not found in the table products.

Nama	Type	Length	Descriptions
idrequest	bigInt	20	
name	varchar	255	
description	text	0	
image	varchar	255	
status	varchar	255	- New - Approve - Reject
created_by	Int	10	
updated_by	int	10	
created_at	timestamp	0	
updated_at	timestamp	0	
deleted_by	int	10	
deleted_at	timestamp	0	

System Flow

This section will discuss about the system flow of the system. Figure 2 shows about the cloud hospital's server. It is shown that all the drug datasets will be linked between i3L and Lira Medika Hospital's dataset where the drug-drug interaction dataset will be stored in i3L's database. First, when the administrator/user check the drug-drug interaction (DDI) in the prescription, he/she will open the website (front-end website) from Lira Medika Hospital and input the list of the drugs, then the system will check the DDI data through API to the i3L website and database. Then, it will return the DDI information to the administrator/user.

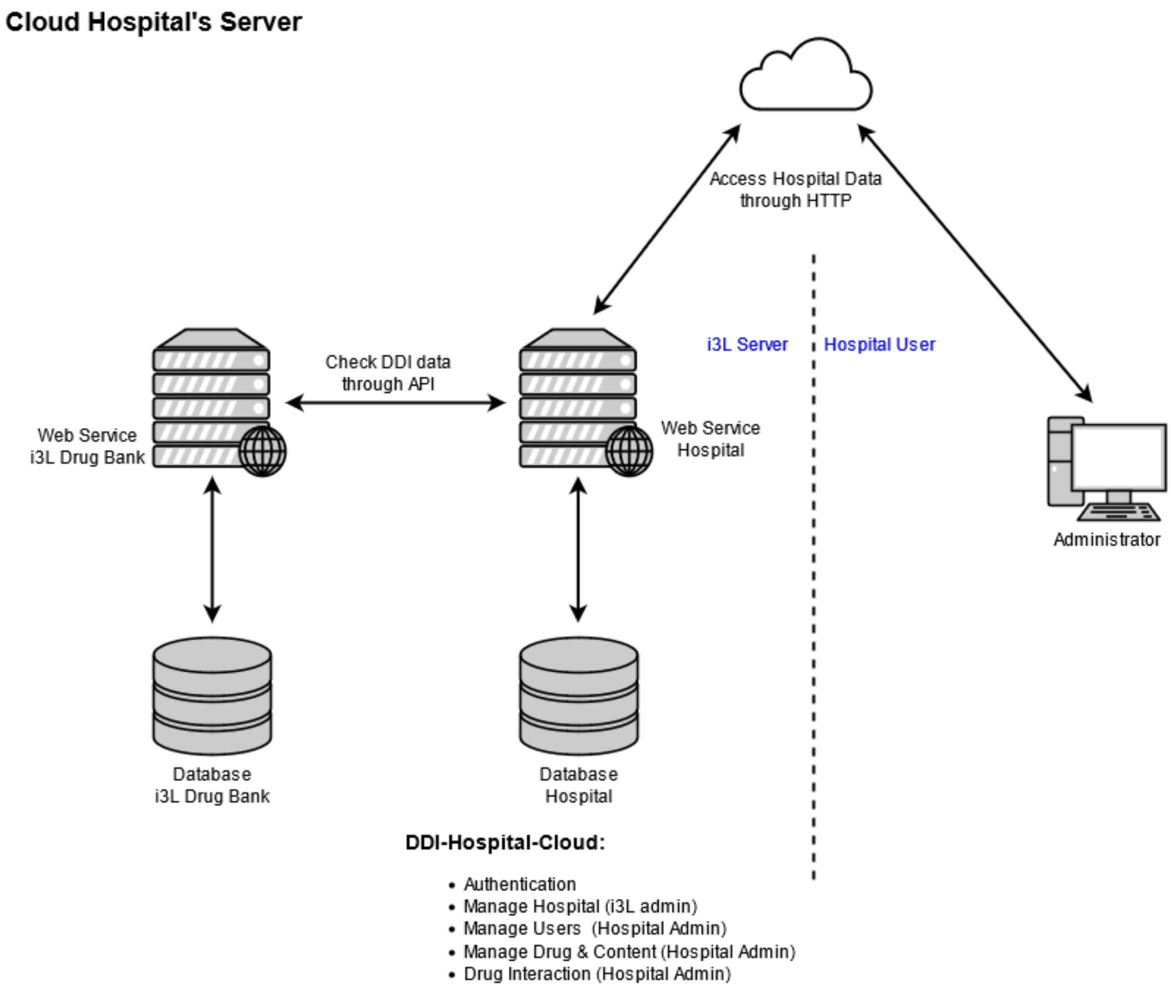


Figure 2. Drug-drug interaction flow.

Flow system for the user can be seen in figure 3. It is shown that the user needs to input the drug names from the prescription to the system. Then, it yields the information if there is and there is no drug interaction occur.

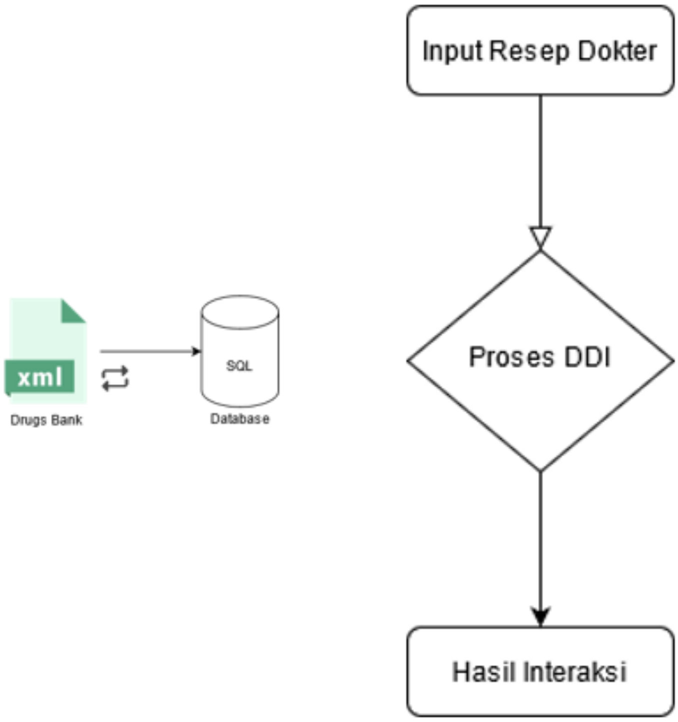


Figure 3. Flow system for the user.

Use case of the system can be seen in the figure 4 where there are administrator and operator. Administrator will contribute to manage the back-end part of the website system where the administrator will manage the information of drug-drug interaction from the drug bank, update the data of the drug-drug interaction and then manage the users. Moreover, the operator will be able to check the DDI from the front-end website.

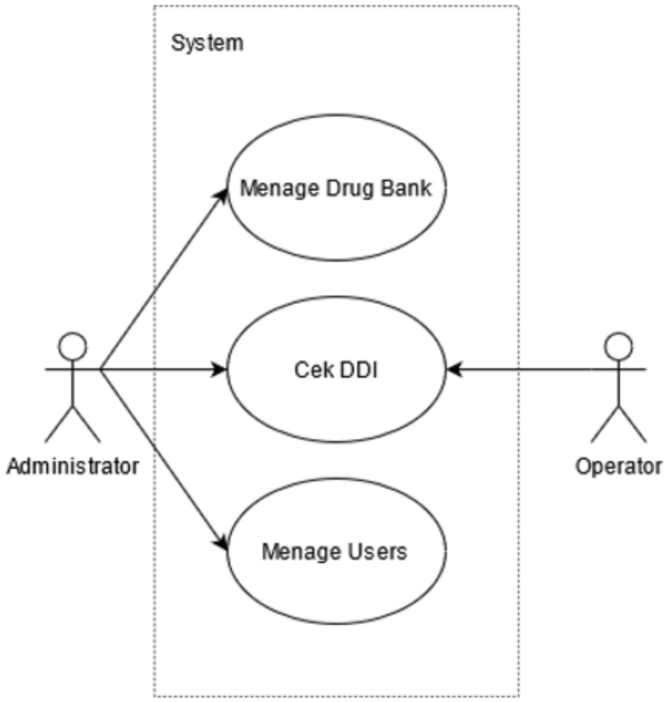


Figure 4. Use Case of the system.

The application flow is shown in the figure 5. The user needs to login in order to input the drugs, then the hospital website will process the dungs and refer it to the i3L website, then as the result it will return the information weather there is drug interaction or not.

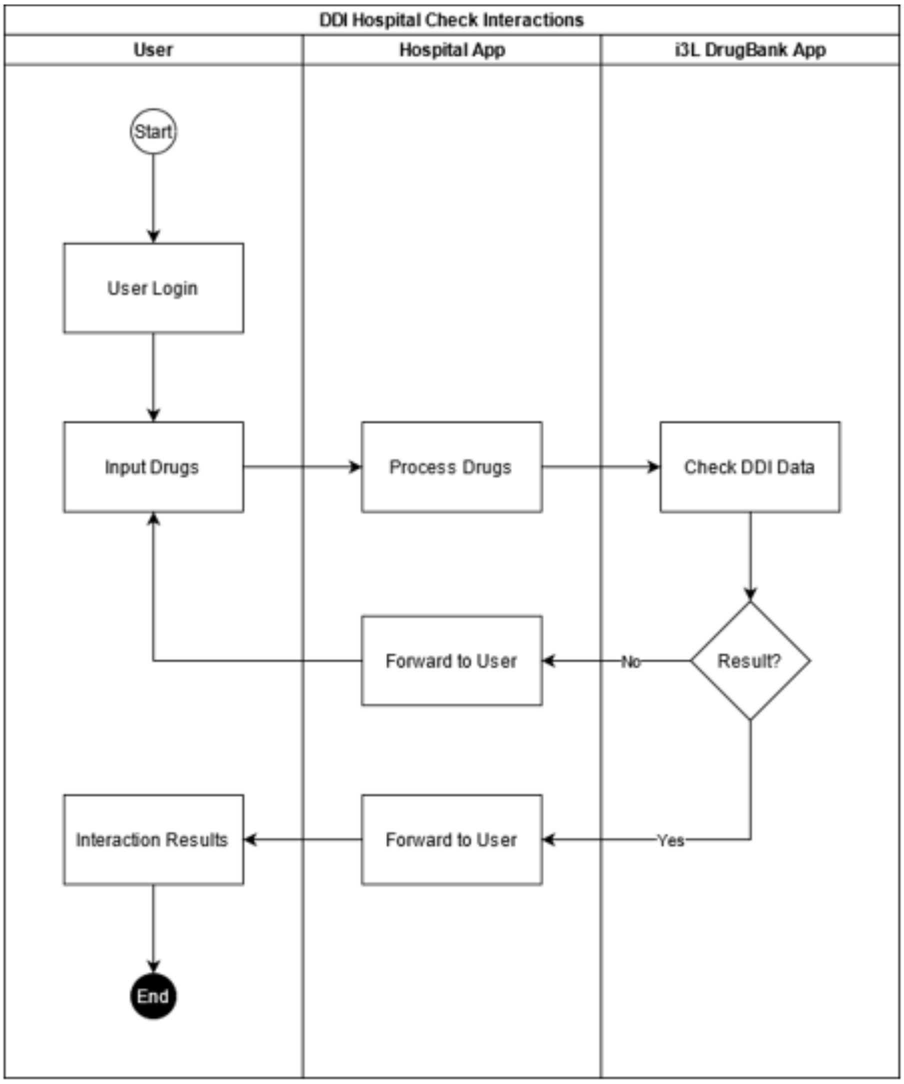


Figure 5. The application flowchart.

Framework

In this section, we will discuss about how to create and integrate the system API. We used a framework Laravel to create a website that store drug-drug interaction datasets. So, the framework is the website that will be connected to all the dataset in the database from the previous section. It can become the interface of the database. Below are step by step to create the website.

1. Install Framework

   This system used Laravel to create the website. All the information related to laravel software can be accessed at: https://laravel.com/docs/8.x/.

2. Setup Configuration

After laravel is installed on your computer, then we need to do the configuration. Below are the list of activities that need to be completed.

a. Create a migration file

File migration is used to create a table and interaction between the tables. The process and the output are shown in figure 6 and figure 7, respectively. The detail explanation can be accessed at: https://laravel.com/docs/8.x/database.

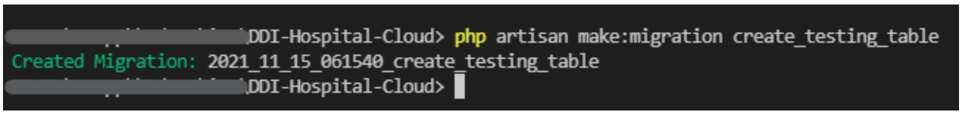


Figure 6. Example of code to create a migration file.

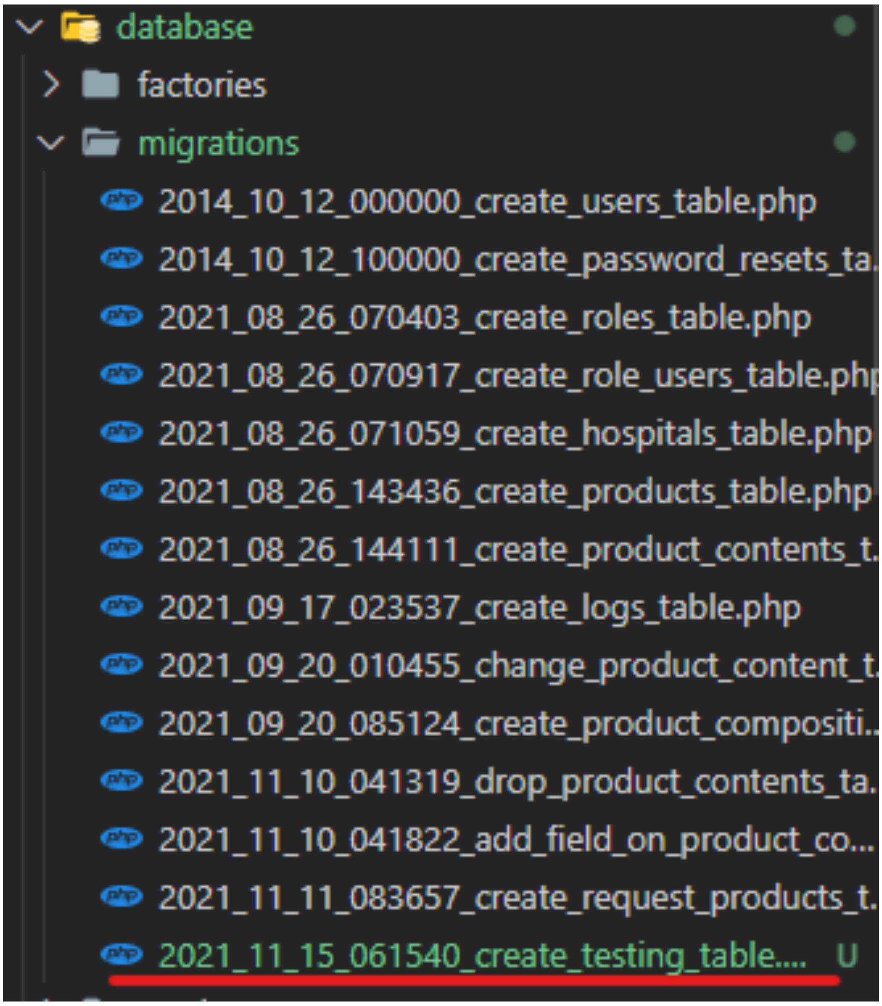


Figure 7. Migration file in the folder.

b. Setup .env file

This setting is used to manage the connection to the database. The configuration or setting can be seen in the highlighted red color of figure 8.

```
.env
APP_NAME=DDI-Hospital-Cloud
APP_ENV=local
APP_KEY=base64:ASquMd0AFKSljSS+QgHAyo45r8Bai4sHnf4s4H6ft5Q=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=daily
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=ddi
DB_USERNAME=root
DB_PASSWORD=

```

Figure 8. The configuration to connect the system to the database.

c. Create Models file https://laravel.com/docs/8.x/eloquent
Laravel use model file to access all the tables from database. The process, the output and part of the coding is shown in figure 9, figure 10 and figure 11, respectively.

```
[redacted]DDI-Hospital-Cloud> php artisan make:model TestingModel
Model created successfully.
[redacted]\DDI-Hospital-Cloud>
```

Figure 9. The process to create a Models

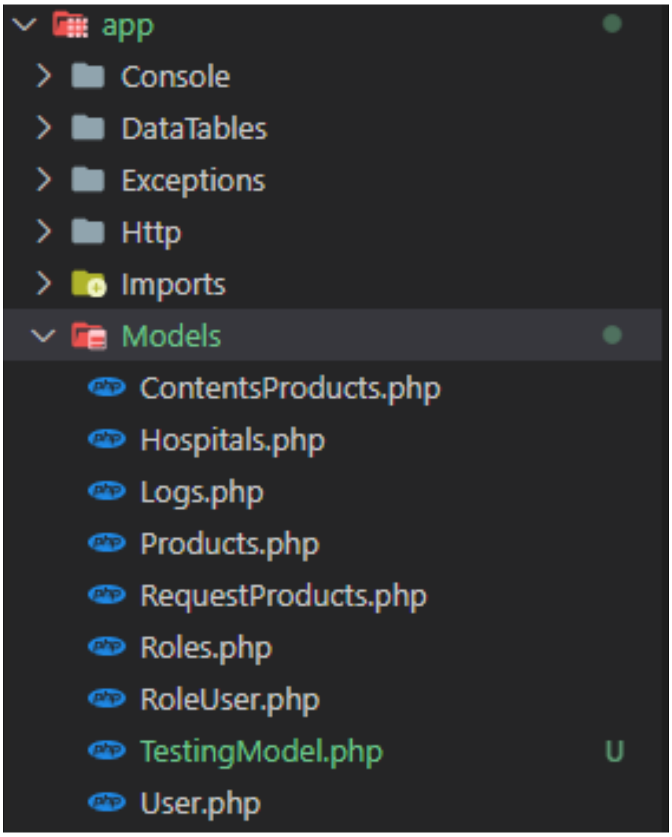


Figure 10. The list of models created for the system.

TestingModel.php U

app > Models > TestingModel.php > ...

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;

class TestingModel extends Model
{
    use HasFactory, SoftDeletes;

    protected $table = 'nama_tabel';

    protected $primaryKey = 'primary_key';

    protected $fillable = [
        'field_1','field_2','field_3'
    ];

    public function createdBy(){
        return $this->belongsTo(User::class, 'created_by', 'idusers')->withTrashed();
    }

    public function updatedBy(){
        return $this->belongsTo(User::class, 'updated_by', 'idusers')->withTrashed();
    }

    public function deletedBy(){
        return $this->belongsTo(User::class, 'deleted_by', 'idusers')->withTrashed();
    }
}
```

Figure 11. The example code of a Model.

d. Create a controllers file https://laravel.com/docs/8.x/controllers

   File Controller is used by Laravel to processes the Logic and bridging between Views and Database. The process and the output can be seen in the figure 12 and figure 13, respectively.

```
[redacted]\DDI-Hospital-Cloud> php artisan make:controller TestingController
Controller created successfully.
[redacted]\DDI-Hospital-Cloud>
```

Figure 12. The process to create controllers.

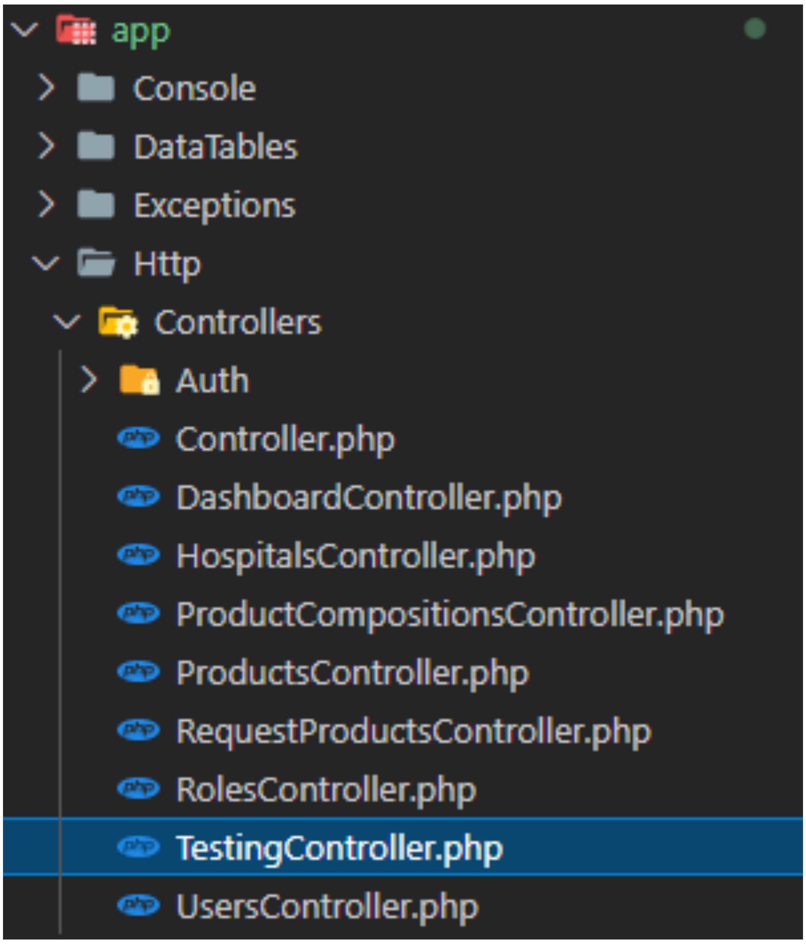


Figure 13. The list of controllers created for the system

e. Register the “Gate” file

The figure 14 shows the code to register the Gate Authorized, it is used to limited *permissions* to the *users.* The detail explanation can be studied at: https://laravel.com/docs/8.x/authorization

```
AuthServiceProvider.php ×
app > Providers > AuthServiceProvider.php > PHP Intelephense > AuthS

     *
     * @return void
     */
    public function boot()
    {
        $this->registerPolicies();

        $this->registerUsersPolicies();
        $this->registerHospitalsPolicies();
        $this->registerProductsPolicies();
        $this->registerProductCompositionsPolicies();
        $this->registerDashboardPolicies();
        $this->registerProductRequest();
        $this->registerTestingPolicies();
    }

    public function registerTestingPolicies(){
        Gate::define('read-testing', function($user){
            return $user->hasAccess(['read-testing']);
        });

        Gate::define('create-testing', function($user){
            return $user->hasAccess(['create-testing']);
        });

        Gate::define('update-testing', function($user){
            return $user->hasAccess(['update-testing']);
        });

        Gate::define('delete-testing', function($user){
            return $user->hasAccess(['delete-testing']);
        });

        Gate::define('status-testing', function($user){
            return $user->hasAccess(['status-testing']);
        });

    }
```

Figure 14. The code to register the gate file

f. Register the “Routes” file https://laravel.com/docs/8.x/routing

The figure below shows the code to arrange Url, Controller, and Middleware (Gate).

web.php

routes > web.php

```php
<?php

use App\Http\Controllers\Auth\LoginController;
use App\Http\Controllers\DashboardController;
use App\Http\Controllers\HospitalsController;
use App\Http\Controllers\ProductCompositionsController;
use App\Http\Controllers\ProductsController;
use App\Http\Controllers\RequestProductsController;
use App\Http\Controllers\RolesController;
use App\Http\Controllers\TestingController;
use App\Http\Controllers\UsersController;
use Illuminate\Support\Facades\Route;

/*
|--------------------------------------------------------------------------
| Web Routes
|--------------------------------------------------------------------------
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/

Route::get('/', [LoginController::class, 'showLoginForm']);
Route::post('logout', [LoginController::class, 'logout'])->name('logout');
Route::post('login', [LoginController::class, 'login'])->name('post.login');

Route::get('testing', [TestingController::class, 'index'])->name('testing.index')->middlewar('can:read-testing');
Route::get('testing/add', [TestingController::class, 'add'])->name('testing.add')->middlewar('can:create-testing');
Route::post('testing', [TestingController::class, 'store'])->name('testing.store')->middlewar('can:create-testing');
```

Figure 15. The code to register the routes file

g. Create Views file

View is the place to design the interface. The figure 16 shows the folder of the view that manage the interface that will be shown to the users. For the detail information can be accessed at: https://laravel.com/docs/8.x/views

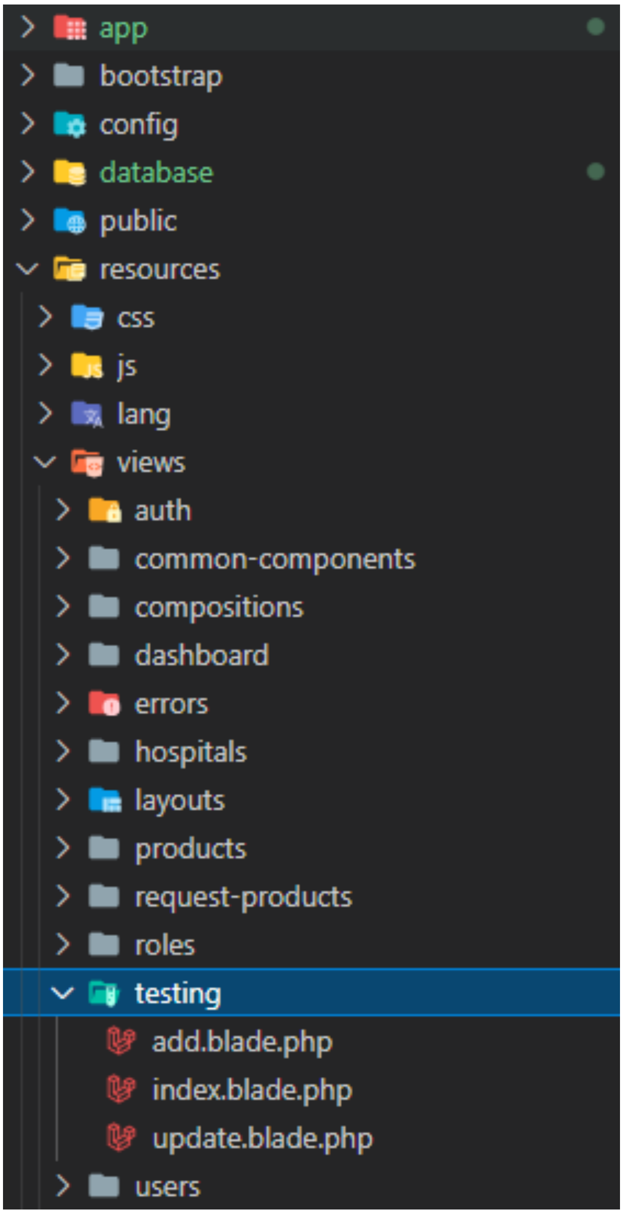
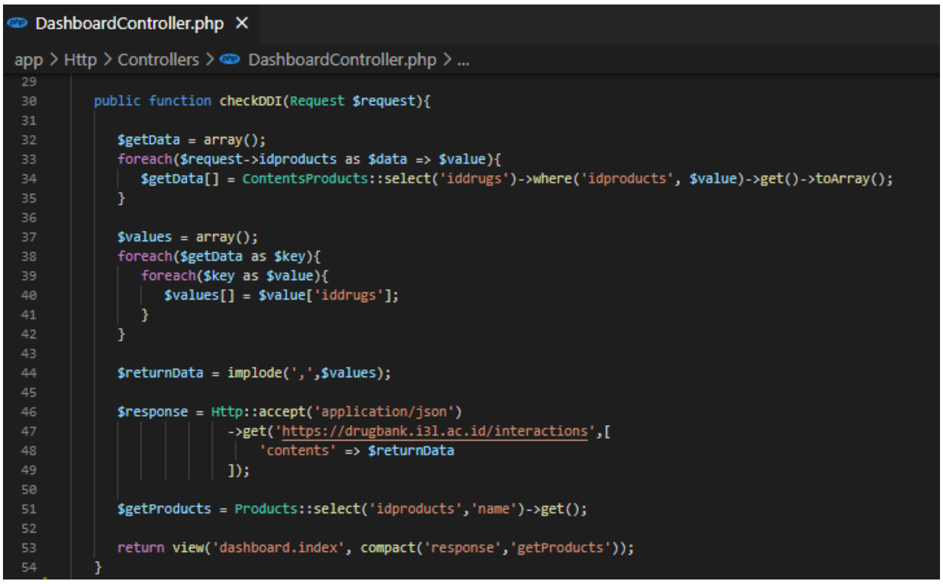


Figure 16. The list of created view

3. API Integration

After the system is created in the previous section, based on figure 2, we need to connect system API between i3L and Lira Medika hospital system. Therefore, once Lira Medika hospital send the request to obtain drug-drug interaction information that stored in i3L's system, then through the API connection the i3L's system will be automatically send the response of the request. Line code number 46 in the figure 17 shows the API integration code. The detail information can be further reviewed at: https://laravel.com/docs/8.x/http-client.




```
29
30      public function checkDDI(Request $request){
31
32          $getData = array();
33          foreach($request->idproducts as $data => $value){
34              $getData[] = ContentsProducts::select('iddrugs')->where('idproducts', $value)->get()->toArray();
35          }
36
37          $values = array();
38          foreach($getData as $key){
39              foreach($key as $value){
40                  $values[] = $value['iddrugs'];
41              }
42          }
43
44          $returnData = implode(',',$values);
45
46          $response = Http::accept('application/json')
47                          ->get('https://drugbank.i3l.ac.id/interactions',[
48                              'contents' => $returnData
49                          ]);
50
51          $getProducts = Products::select('idproducts','name')->get();
52
53          return view('dashboard.index', compact('response','getProducts'));
54      }
```

Figure 17. The code to integrate the API.

References

Assiri, G. A., Shebl, N. A., Mahmoud, M. A., Aloudah, N., Grant, E., Aljadhey, H., & Sheikh, A. (2018). What is the epidemiology of medication errors, error-related adverse events and risk factors for errors in adults managed in community care contexts? A systematic review of the international literature. *BMJ Open*, *8*(5). https://doi.org/10.1136/bmjopen-2017-019101

Avery, T., Barber, N., Ghaleb, B., Franklin, B. D., Armstrong, S., Crowe, S., … Olanrenwaju, T. (2012). Investigating the prevalence and causes of prescribing errors in general practice : The PRACtICe Study (PRevalence And Causes of prescrIbing errors in general practiCe) A report for the GMC. *General Medical Council*, (May), 1–187. Retrieved from http://www.gmc-uk.org/about/research/12996.asp

Bates, D. W., Cullen, D. J., Laird, N., Petersen, L. A., Small, S. D., Servi, D., … Leape, L. L. (1995). Incidence of Adverse Drug Events and Potential Adverse Drug Events: Implications for Prevention. *JAMA: The Journal of the American Medical Association*, *274*(1), 29–34. https://doi.org/10.1001/jama.1995.03530010043033

Cascorbi, I. (2012). Drug Interactions-Principles, Examples, and Clinical Consequences. *Deutsches Arzteblatt International*, *109*(33–34), 546–556. https://doi.org/10.3238/arztebl.2012.0546

Handayani, T. W. (2017). Faktor Penyebab Medication Error di RSU Anutapura Kota Palu. *Perspektif: Jurnal Pengembangan Sumber Daya Insani Vol.02 Nomor 02 Juli-Desember 2017*, *2*(2), 224–229. Retrieved from https://journal.unismuh.ac.id/index.php/Perspektif/article/view/1285

National Coordinating Council for Medication Error Reporting and Prevention. (2015). What is a medication error?

Nurfitria, R. S., Adillah Effendi, R. N., & Iskandar, D. (2019). Potential Medication Errors in Electronic Prescribing in A Primary Health Care. *Indonesian Journal of Pharmaceutical and Clinical Research*, *2*(1), 45–54. https://doi.org/10.32734/idjpcr.v2i1.1166

Palleria, C., Di Paolo, A., Giofrè, C., Caglioti, C., Leuzzi, G., Siniscalchi, A., … Gallelli, L. (2013). Pharmacokinetic drug-drug interaction and their implication in clinical management. *Journal of Research in Medical Sciences*, *18*(7), 600–609.

Sears, K., Ross-White, A., & Godfrey, C. M. (2012). The incidence, prevalence and contributing factors associated with the occurrence of medication errors for children and adults in the community setting: a systematic review. *JBI Library of Systematic Reviews*, *10*(35), 2350–2464. https://doi.org/10.11124/jbisrir-2012-35

World Health Organization. (2016). *Medication Errors: Technical Series on Safer Primary Care*. Geneva: World Health Organization.

Zavaleta-Bustos, M., Castro-Pastrana, L. I., Reyes-Hernández, I., López-Luna, M. A., & Bermúdez-Camps, I. B. (2008). Prescription errors in a primary care university unit: Urgency of pharmaceutical care in Mexico. *Revista Brasileira de Ciencias Farmaceuticas/Brazilian Journal of Pharmaceutical Sciences*, *44*(1), 115–125. https://doi.org/10.1590/S1516-93322008000100013

i3L
PRESS